Automating Production of Cross Media Content for Multi-channel Distribution

www.AXMEDIS.org

DE3.1.2.3.7
Specification of AXMEDIS External Processing Algorithms, update of DE3.1.2.2.7

Version: 1.0
Date: 17-07-2007
Responsible: Peter Ebinger (FHGIGD) (revised and approved by coordinator)

Project Number: IST-2-511299 Project Title: AXMEDIS Deliverable Type: report Visible to User Groups: yes Visible to Affiliated: yes Visible to the Public: yes
Deliverable Number: DE3.1.2.3.7 Contractual Date of Delivery: Actual Date of Delivery: 17/07/2007 Title of Deliverable: Specification of AXMEDIS External Processing Algorithms Work-Package contributing to the Deliverable: WP3.1 Task contributing to the Deliverable: WP3, WP2 Nature of the Deliverable: report Author(s): FHGIGD, EPFL, DSI, DIPITA, UNIVLEEDS, UPC, UR

Abstract: this part includes the specification of components, formats, databases and protocol related to the AXMEDIS Framework area Content Processing including algorithms of adaptation, fingerprint, watermark, etc., of documents, video, images, audio files, multimedia, metadata, licenses, etc.

Keyword List: AXMEDIS content processing, adaptation, watermark, fingerprint, descriptors

AXMEDIS Copyright Notice

The following terms (including future possible amendments) set out the rights and obligations licensee will be requested to accept on entering into possession of any official AXMEDIS document either by downloading it from the web site or by any other means.

Any relevant AXMEDIS document includes this license. PLEASE READ THE FOLLOWING TERMS CAREFULLY AS THEY HAVE TO BE ACCEPTED PRIOR TO READING/USE OF THE DOCUMENT.

1. DEFINITIONS

i. "**Acceptance Date**" is the date on which these terms and conditions for entering into possession of the document have been accepted.

ii. "**Copyright**" stands for any content, document or portion of it that is covered by the copyright disclaimer in a Document.

iii. "**Licensor**" is AXMEDIS Consortium as a de-facto consortium of the EC project and any of its derivations in terms of companies and/or associations, see www.axmedis.org

iv. "**Document**" means the information contained in any electronic file, which has been published by the Licensor's as AXMEDIS official document and listed in the web site mentioned above or available by any other means.

v. "**Works**" means any works created by the licensee, which reproduce a Document or any of its part.

2. **LICENCE**

1. The Licensor grants a non-exclusive royalty free licence to reproduce and use the Documents subject to present terms and conditions (the **Licence**) for the parts that are own and proprietary property the of AXMEDIS consortium or its members.
2. In consideration of the Licensor granting the Licence, licensee agrees to adhere to the following terms and conditions.

3. **TERM AND TERMINATION**

1. Granted Licence shall commence on Acceptance Date.
2. Granted Licence will terminate automatically if licensee fails to comply with any of the terms and conditions of this Licence.
3. Termination of this Licence does not affect either party's accrued rights and obligations as at the date of termination.
4. Upon termination of this Licence for whatever reason, licensee shall cease to make any use of the accessed Copyright.
5. All provisions of this Licence, which are necessary for the interpretation or enforcement of a party's rights or obligations, shall survive termination of this Licence and shall continue in full force and effect.
6. Notwithstanding License termination, confidentiality clauses related to any content, document or part of it as stated in the document itself will remain in force for a period of 5 years after license issue date or the period stated in the document whichever is the longer.

4. **USE**

1. Licensee shall not breach or denigrate the integrity of the Copyright Notice and in particular shall not:
   i. remove this Copyright Notice on a Document or any of its reproduction in any form in which those may be achieved;
   ii. change or remove the title of a Document;
   iii. use all or any part of a Document as part of a specification or standard not emanating from the Licensor without the prior written consent of the Licensor; or
   iv. do or permit others to do any act or omission in relation to a Document which is contrary to the rights and obligations as stated in the present license and agreed with the Licensor

1. **COPYRIGHT NOTICES**

    1. All Works shall bear a clear notice asserting the Licensor's Copyright. The notice shall use the wording employed by the Licensor in its own copyright notice unless the Licensor otherwise instructs licensees.

2. **WARRANTY**

    1. The Licensor warrants the licensee that the present licence is issued on the basis of full Copyright ownership or re-licensing agreements granting the Licensor full licensing and enforcement power.
    2. For the avoidance of doubt the licensee should be aware that although the Copyright in the documents is given under warranty this warranty does not extend to the content of any document which may contain references or specifications or technologies that are covered by patents (also of third parties) or that refer to other standards. AXMEDIS is not responsible and does not guarantee that the information contained in the document is fully proprietary of AXMEDIS consortium and/or partners.
    3. Licensee hereby undertakes to the Licensor that he will, without prejudice to any other right of action which the Licensor may have, at all times keep the Licensor fully and effectively indemnified against all and any liability (which liability shall include, without limitation, all losses, costs, claims, expenses, demands, actions, damages, legal and other professional fees and expenses on a full indemnity basis) which the Licensor may suffer or incur as a result of, or by reason of, any breach or non-fulfillment of any of his obligations in respect of this License.

3. **INFRINGEMENT**

    1. Licensee undertakes to notify promptly the Licensor of any threatened or actual infringement of the Copyright which comes to licensee notice and shall, at the Licensor's request and expense, do all such things as are reasonably necessary to defend and enforce the Licensor's rights in the Copyright.

4. **GOVERNING LAW AND JURISDICTION**

    1. This Licence shall be subject to, and construed and interpreted in accordance with Italian law.
    2. The parties irrevocably submit to the exclusive jurisdiction of the Italian Courts.

Please note that:

- You can become affiliated with AXMEDIS. This will give you the access to a huge amount of knowledge, information and source code related to the AXMEDIS Framework. If you are interested please contact P. Nesi at nesi@dsi.unifi.it. Once affiliated with AXMEDIS you will have the possibility of using the AXMEDIS specification and technology for your business.
- You can contribute to the improvement of AXMEDIS documents and specification by sending the contribution to P. Nesi at nesi@dsi.unifi.it
- You can attend AXMEDIS meetings that are open to public, for additional information see http://www.axmedis.org or contact P. Nesi at nesi@dsi.unifi.it

Table of Content

1 Executive Summary and Report Scope

The full AXMEDIS specification document has been decomposed in the following parts:

DE number	Deliverable title	responsible
DE3.1.2.3.1	Specification of General Aspects of AXMEDIS framework AXMEDIS-DE3-1-2-3-1-Spec-of-AX-Gen-Asp-of-AXMEDIS-framework	DSI
DE3.1.2.3.2	Specification of AXMEDIS Command Manager AXMEDIS- DE3-1-2-3-2-Spec-of-AX-Cmd-Man	DSI
DE3.1.2.3.3	Specification of AXMEDIS Object Manager and Protection Processor AXMEDIS-DE3-1-2-3-3-Spec-of-AXOM-and-ProtProc	DSI
DE3.1.2.3.4	Specification of AXMEDIS Editors and Viewers AXMEDIS-DE3-1-2-3-4-Spec-of-AX-Editors-and-Viewers	DSI
DE3.1.2.3.5	Specification of External AXMEDIS Editors/Viewers and Players AXMEDIS-DE3-1-2-3-5-Spec-of-External-Editors-Viewers-Players	DSI
DE3.1.2.3.6	Specification of AXMEDIS Content Processing AXMEDIS-DE3-1-2-3-6-Spec-of-AX-Content-Processing	DSI
DE3.1.2.3.7	Specification of AXMEDIS External Processing Algorithms AXMEDIS-DE3-1-2-3-7-Spec-of-AX-External-Processing-Algorithms	FHGIGD
DE3.1.2.3.8	Specification of AXMEDIS CMS Crawling Capabilities AXMEDIS-DE3-1-2-3-8-Spec-of-AX-CMS-Crawling-Capab	DSI
DE3.1.2.3.9	Specification of AXMEDIS database and query support AXMEDIS-DE3-1-2-3-9-Spec-of-AX-database-and-query-support	EXITECH
DE3.1.2.3.10	Specification of AXMEDIS P2P tools, AXEPTool and AXMEDIS AXMEDIS-DE3-1-2-3-10-Spec-of-AXEPTool-and-AXMEDIA-tools	DSI
DE3.1.2.3.11	Specification of AXMEDIS Programme and Publication tools AXMEDIS-DE3-1-2-3-11-Spec-of-AX-Progr-and-Pub-tool	UNIVLEEDS
DE3.1.2.3.12	Specification of AXMEDIS Workflow Tools AXMEDIS-DE3-1-2-3-12-Spec-of-AX-Workflow-Tools	UR
DE3.1.2.3.13	Specification of AXMEDIS Certifier and Supervisor and networks of AXCS AXMEDIS-DE3-1-2-3-13-Spec-of-AXCS-and-networks	DSI
DE3.1.2.3.14	Specification of AXMEDIS Protection Support AXMEDIS-DE3-1-2-3-14-Spec-of-AX-Protection-Support	UPC
DE3.1.2.3.15	Specification of AXMEDIS accounting and reporting AXMEDIS-DE3-1-2-3-15-Spec-of-AX-Accounting-and-Reporting	EXITECH

1.1 This document concerns

This document AXMEDIS-DE3-1-2-2-7 concerns the specification of the AXMEDIS External Processing. It is an updated version of part D of DE3.1.2. As a consequence of the update process, some parts originally available in DE3.1.2D have been moved to other deliverables and the focus of this document changed on external processing algorithms as reflected in the new title.

The external processing algorithms are grouped according to the functionality class they belong to:

- **Content adaptation** tools and algorithms are the algorithms that allow automatic content processing. The input and the output content type are the same. E.g. typical applications are scenarios where content is adapted according to the needs of the receiver.
- **Content description** tools and algorithms automatically extract low or high level descriptors from content. AXMEDIS supports the overall range of the content descriptors through is general description.
- **Content fingerprinting** tools and algorithms are a subset of low level descriptors. Their purpose is the identification and authentication of content.
- **External processing** algorithms are external algorithms not belonging to the above group that are so far considered to be integrated within the AXMEDIS framework.

1.2 List of Modules or Executable Tools Specified in this document

A module is a component that can be or it is reused in other cases or points of the AXMEDIS framework or of other AXMEDIS based solutions.
The modules/tools have to include effective components and/or tools and also testing components and tools.

Module/tool Name	Module/Tool Description and purpose, state also in which other AXMEDIS area is used	Standards exploited if any
Adaptation Tools And Algorithms for Text	This external library provides the functionality to adapt and reformat the input text according to the needs of the specific customer and scenario.	
Adaptation Tools And Algorithms for Video	This external library provides the functionality to adapt and reformat the input video according to the needs of the specific customer and scenario.	
Adaptation Tools And Algorithms for Images	This external library provides the functionality to adapt and reformat the input image according to the needs of the specific customer and scenario.	
Adaptation Tools And Algorithms for Audio	This external library provides the functionality to adapt and reformat the input audio according to the needs of the specific customer and scenario	
Adaptation Tools And Algorithms for Multimedia	This external library provides the functionality to adapt and reformat the input multimedia scenes according to the needs of the specific customer and scenario	
Adaptation Tools And Algorithms for Meta Data	Tools/Libraries for Metadata adaptation. Used by Metadata Mapper GUI and JS_Metadata Javascript wrapper	XSLT
Adaptation Tools And Algorithms for DRM Information	This external library provides the functionality to adapt the input text DRm information according to the needs of the specific customer and scenario	
Adaptation Tools And Algorithms for Ringtones	This external library provides the functionality to adapt the ring tones on the fly, to various formats depending on the needs of the customer.	Adaptation Tools And Algorithms for ring tones
Descriptor Extractor as Fingerprint for Text Files	This external library provides the functionality to extract content descriptors for text.	MPEG-7
Descriptor Extractor as Fingerprint for Audio Files	This external library provides the functionality to extract content descriptors of audio to be used for content based retrieval.	MPEG-7

Descriptor Extractor as Fingerprint for Video Files	This external library integrates functionality provided in the reference implementation of MPEG-7.	MPEG-7
Fingerprint Estimation for Text Files	This external library provides the functionality to authenticate text documents.	MPEG-7
Fingerprint Estimation for Audio Files	This external library provides the functionality to authenticate and to identify audio content. For the input a perceptual hash is calculated, which is robust against processing operations and allows the reliable content authentication and identification.	MPEG-7
Fingerprint Estimation for Video Files	This external library provides the functionality to authenticate and to identify video content. For the input a perceptual hash is calculated, which is robust against processing operations and allows the reliable content authentication and identification.	MPEG-7
Fingerprint Estimation for Metadata	This external library provides the functionality to authenticate and to identify video content. For the input a cryptographic hash is calculated.	MPEG-7
Fingerprint Estimation for Generic Files	This external library provides the functionality to authenticate and to identify video content. For the input a cryptographic hash is calculated.	MPEG-7
Watermarking Audio Files	This external library provides the functionality to read and embed an audio watermark into the audio files.	
External Protection Libraries	This external library provides the functionalities for cryptographic operations on the content file or on information related to it. As specified in the AXMEDIS frameworks will be used as security processor. Input depends on the content validation requirements.	standard encryption algorithms

1.3 List of Formats Specified in this document

A format can be

(i) an XML content file for modelling some information,
(ii) a file format for storing information,
(iii) a format that is manipulated by the tools described in this document, etc...

Format Name	Format Description and purpose, state also in which other modules is used	Standards exploited if any
Metadata Map Format	Format is used to store rules for transforming between Metadata languages	XSLT
Descriptor Formats	Descriptors are stored in XML in the MPEG-7 format	XSLT, MPEG-7
Fingerprint Formats	Descriptors are stored in XML in the MPEG-7 format	XSLT, MPEG-7

2 General Use Cases and scenarios

Access to the content external processing algorithms within the AXMEDIS framework is potentially possible from all tools as these algorithms are accessed via the AXOM. So far, the AXMEDIS framework foresees and supports the following use cases:

1. **Content Crawling**: calling external databases and transferring content into an AXMEDIS database
2. **AXMEDIS Object Editing**: using the AXMEDIS editor
3. **Automatic Content Processing**: using the AXCPEngine
4. **Receiving Content from the AXEPTool**: content that is received via the AXEPTool is automatically authenticated

2.1 Use Case ”Content Crawling”

Figure: Estimating Fingerprint: Collector Engine

The collector engine can call the available plug-ins for the automatic generation of meta-data and content descriptions for as well as for automatic processing of the content the is inserted into the AXMEDIS database.

2.2 Use Case ”AXMEDIS Object Editing”

Figure: Estimating Fingerprint: AXMEDIS Editor

The external processing algorithms are managed by the content processing plug-in manager. It is accessible vie the AXOM. The AXMEDIS Object Editor and Viewer can access all available plug-ins through the AXOM.

2.3 Use Case ”Automatic Content Processing”

Details on this use case are given in DE3-1-2-2-6 (Content Processing)

2.4 Use Case ”Receiving Content from the AXEPTool”

Figure: Using fingerprint for verification of object consistency

The upload content can be processed by the external processing algorithms via the AXOM.

3 General architecture and relationships among the modules produced

In this section, tools and algorithms that will be used in the content adaptation task are described. According to the UML diagram reported below, several content adaptation modules will be developed in order to cope with different types of contents, in particular: video, audio, document, multimedia, DRM, meta data. Each of them will be based on existing library, executable or others. Other and specific algorithms could be added during the life of project.

1. The **AXMEDIS Object Manager** manages AXMEDIS objects. For example, it provides access to to functions and methods for manipulating and managing resources and meta data,

1. The **AXMEDIS Content Processing** is based on a PlugIn Manager and the External Procedures Profile Manager. It allows a flexible and dynamic extension of the functionality available in the AXMEDIS framework by providing a interface to the available Plug-ins

2. **AXMEDIS PlugIns** are a collection of algorithms and tools that provide functions to support a broad variety of applications. The role of the individual components is to provided different methods to process digital contents in order to satisfy several and different user profile.

    a) **Adaption algorithms** provide the functionality for content processing, manipulation and adaptation.

    b) **Description algorithms** provide the functionality for the calculation of high and low level content descriptors for/of digital content.

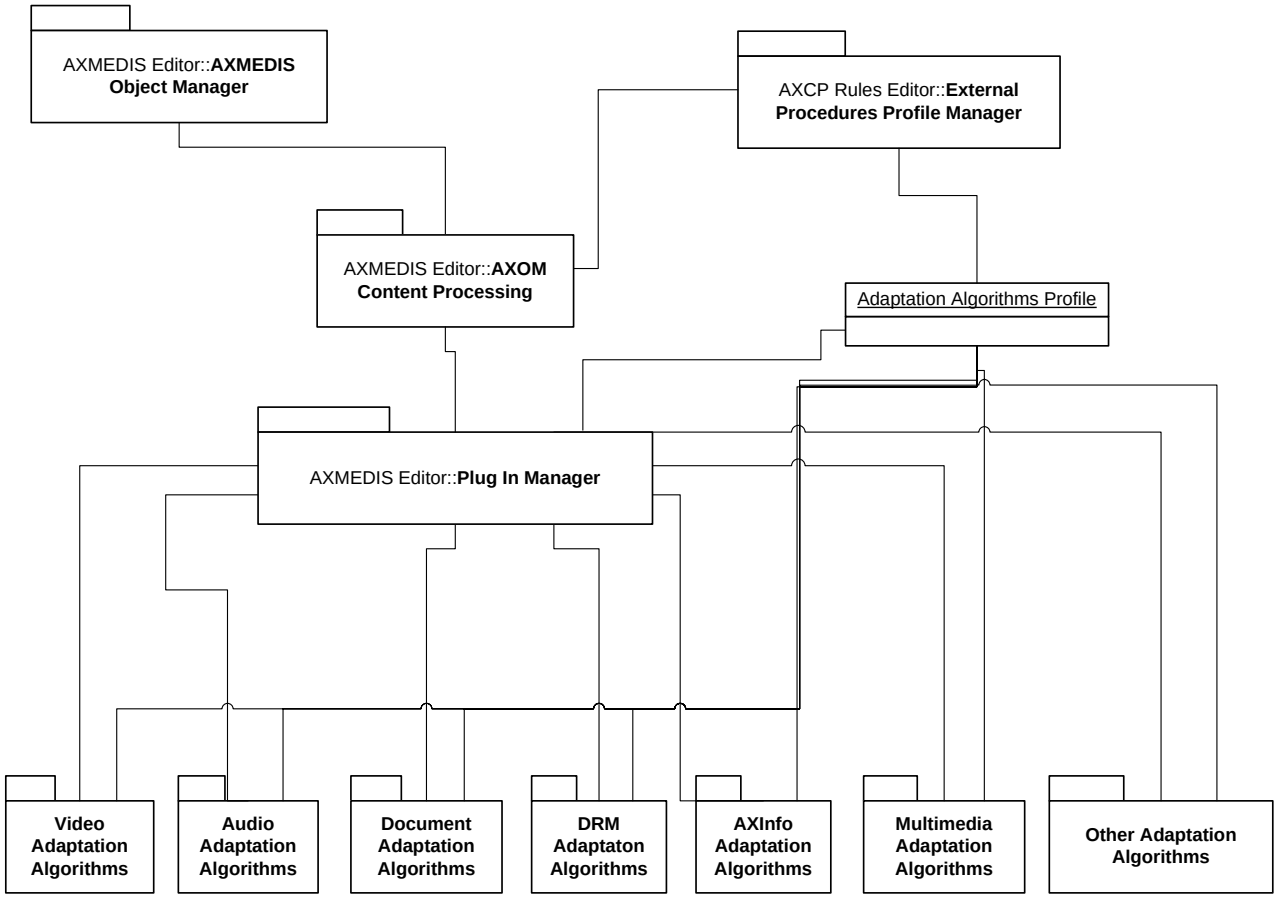
    c) **Fingerprinting algorithms** provide the functionality for the calculation of digital fingerprint and perceptual hashes.

    d) **Watermarking algorithms** provide the functionality for reading and embedding (copyright) information directly in content.

    e) **External protection algorithms** provide the functionality for protecting digital content.

These algorithm classes are the ones initially integrated in AXMEDIS framework. Further classes of functionality can be provided by including existing tools via the flexible interface.

Adaptation Tool and Algorithms



4 Adaptation Tools and Algorithms for Text (DIPITA)

Module/Tool Profile Tools and Algorithms for Document Adaptation		
Responsible Name	Fabbri	
Responsible Partner	DIPITA	
Status (proposed/approved)	Proposed	
Implemented/not implemented	implemented	
Status of the implementation	45%	
Executable or Library/module (Support)	Library	
Single Thread or Multithread	Single thread	
Language of Development	C++	
Platforms supported	MS WINDOWS	
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/newrepos/Framework/{source,include,project}/adaptation/document/	
Reference to the AXFW location of the demonstrator executable tool for internal download		
Reference to the AXFW location of the demonstrator executable tool for public download		
Address for accessing to WebServices if any, add accession information (user aNd Passwd) if any		
Test cases (present/absent)	absent	
Test cases location		
Usage of the AXMEDIS configuration manager (yes/no)	no	
Usage of the AXMEDIS Error Manager (yes/no)	no	
Major Problems not solved	GPL libraries exploitation and distribution.	
Major pending requirements	only pdf to txt and html to txt exploited	
Interfaces API with other tools, named as	Name of the communicating tools References to other major components needed	Communication model and format (protected or not, etc.)
Formats Used	Shared with	format name or reference to a section
PDF		
HTML		
RTF		
Plain text		
Postscript		
Protocol Used	Shared with	Protocol name or reference to a

		section
Used Database name		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
DOCFRAC	3.1.1	LGPL
GNU Ghostscript	8.15	GPL
XPDF	3.0.0	GPL
HTMLDOC	1.8.24rc1	GPL

4.1.1 General Description of the Module

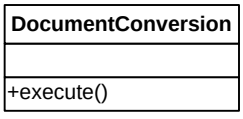
Document adaptation tools provide functions which can convert a text document file modifying its format. The module exploits libraries distributed under GPL, so the conversion part of the tool will be developed as a separated executable program: this part (that will be distributed under GPL) will contain code that will use those libraries while the plug-in will make calls to the converter and won't make direct calls to the libraries so it has not to be licensed under GPL.

The communication protocol between the two parts of the tool has to be defined and provided by DSI.

The plug-in doesn't need to be configured. Our plug-in cannot be produced re-using other partners' components, but it can be used in all plug-in based application such as AXEditor or AXRuleEditor.

This plug-in is designed to be executed only on MS Windows platforms. Information printing capability depends on the application that uses the plug-in. So far, help messages are provided in English only.

API documentation has been generated using Doxygen (http://www.stack.nl/~dimitri/doxygen/) as automatic documentation generating tool and has been posted on the AXMEDIS CVS repository (https://cvs.axmedis.org/newrepos/Framework/doc/code/adaptation/document/html/).



4.1.2 Technical and Installation information

The plug-in is designed to run on Microsoft Windows 2000/NT/XP platforms (it has been tested on Microsoft Windows XP professional and Microsoft Windows 2000). At least a Pentium III processor and 256MB RAM are needed. 6MB of disk space is needed.

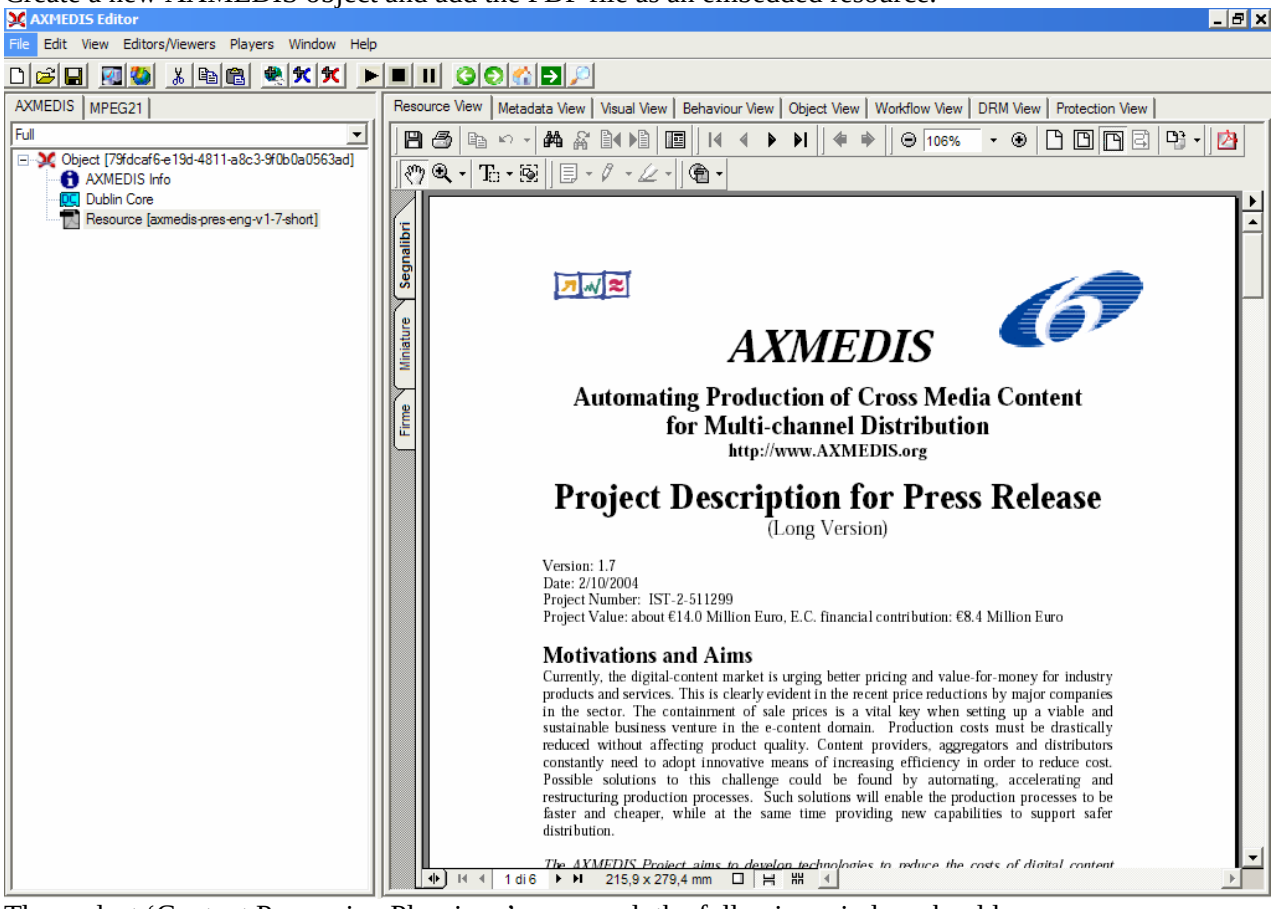
References to other major components needed	
Problems not solved	
Configuration and execution context	

4.1.3 Draft User Manual and examples of usage

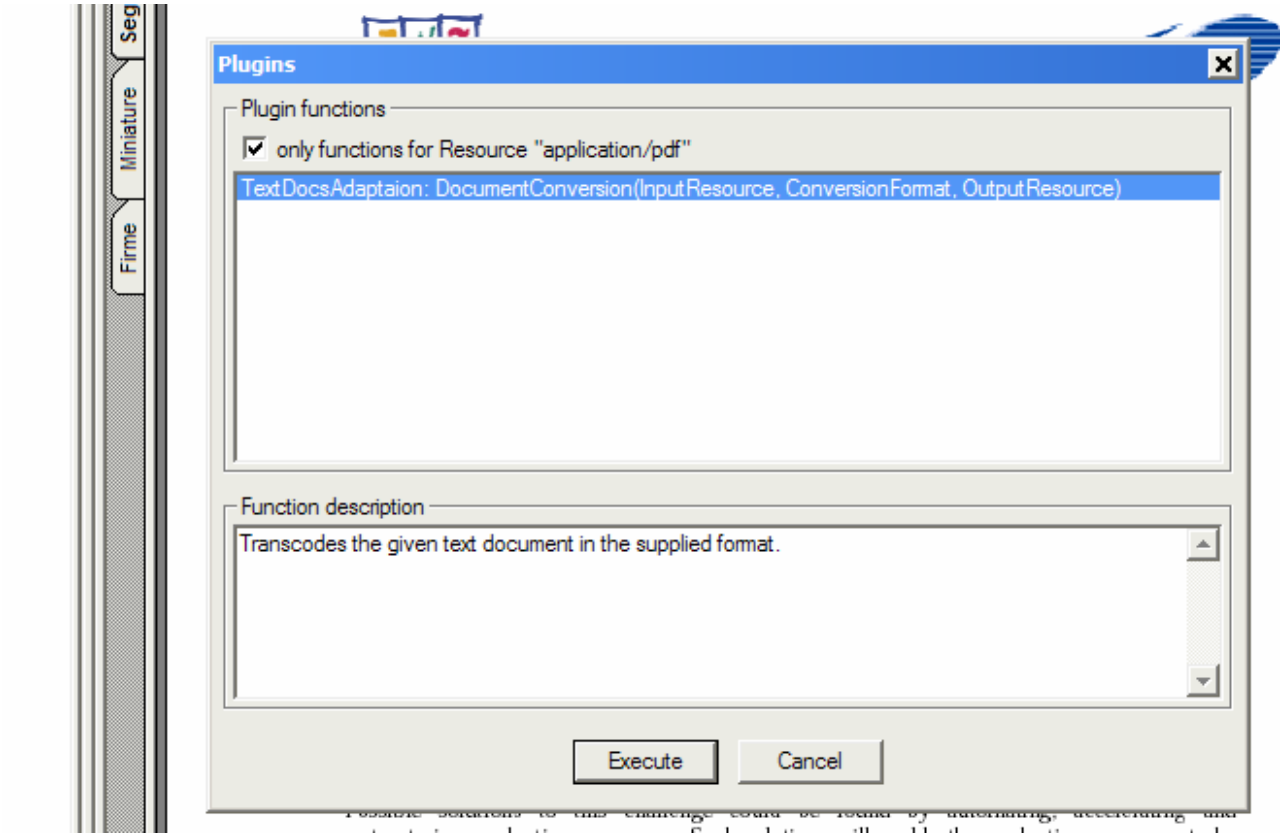
Below, an example on how to use the plug-in with AXEditor.

So far, the plug-in can be applied only to PDF and HTML resources and the output will be text/plain only. In the demonstrator package there is a sample PDF file to test: axmedis-pres-eng-v1-7-short.pdf.

Create a new AXMEDIS object and add the PDF file as an embedded resource.

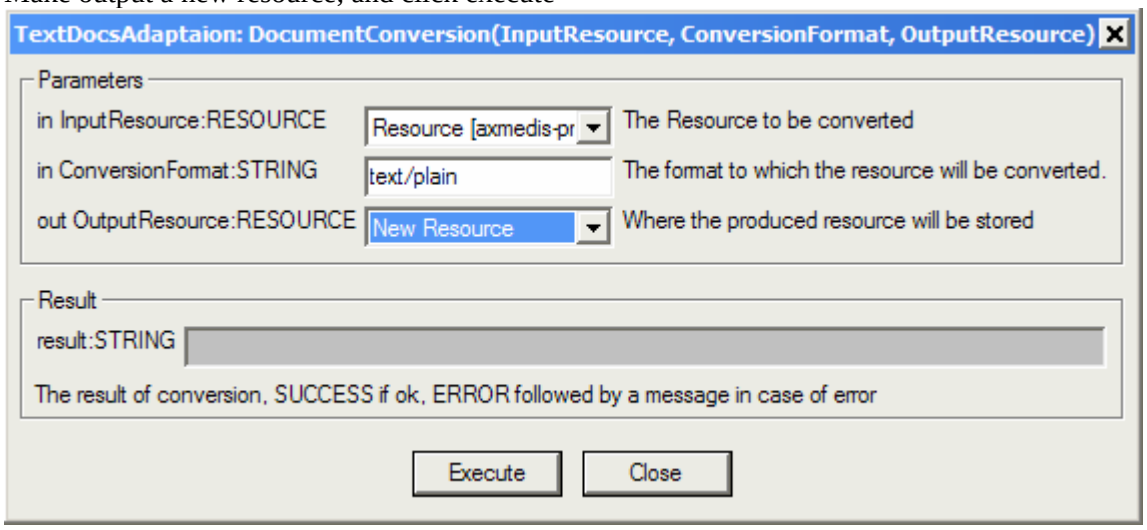


Then select ‘Content Processing Plug-in…’ command; the following window should appear:

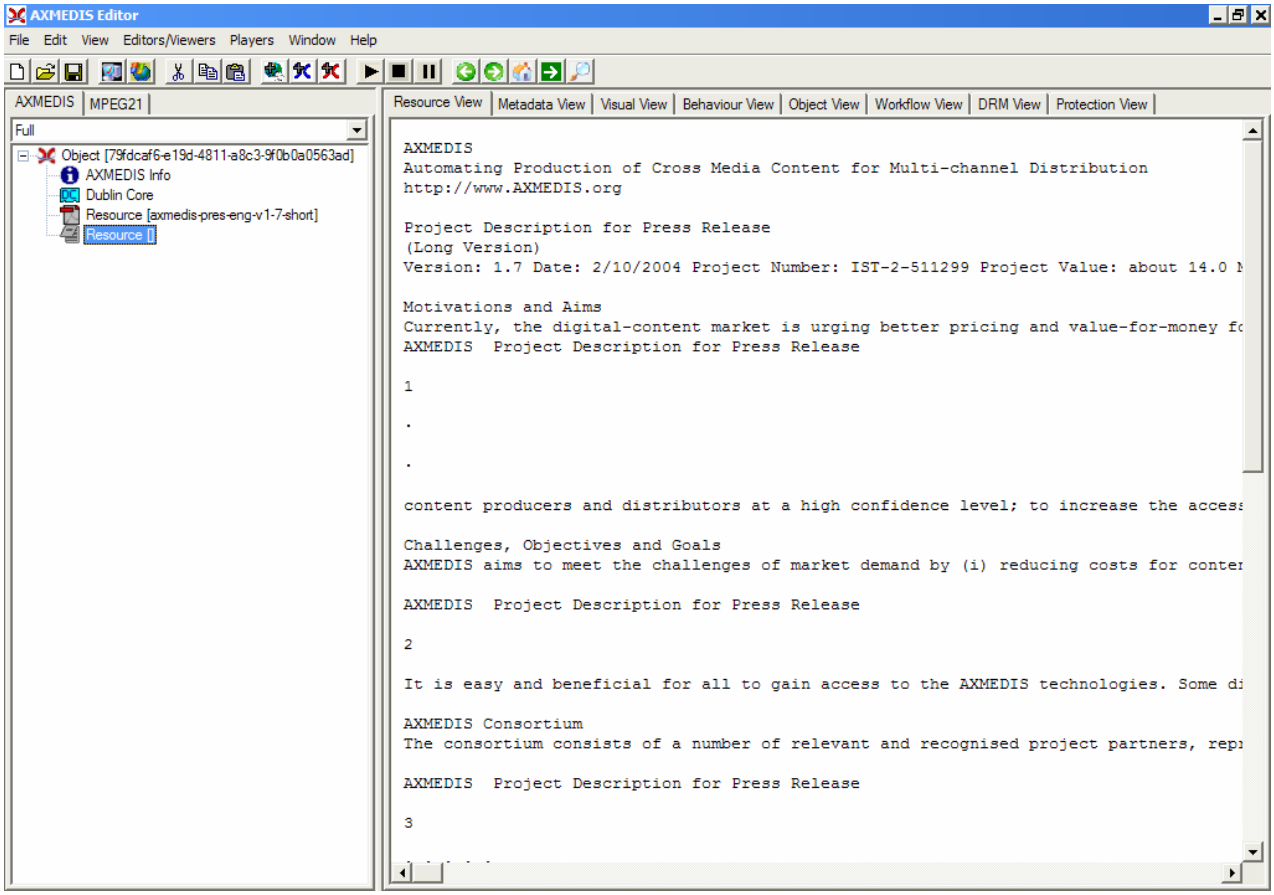


There is only one function available.
DocumentConversion: it will make the transcoding to the format specified as the requested parameter.

The only accepted value for the parameter is: text/plain
Make output a new resource, and click execute



Here's the plain text version of the file:



4.1.4 Formal description of algorithm

The plug-in simply calls exploits the right external library method deending on the input and output document formats.

name	
Method	DocumentConversion
Description	This is the method to call to convert a text document resource
Input parameters	RESOURCE: InputResource – The resource to be converted STRING: ConversionFormat – The desired output format of the text document
Output parameters	RESOURCE: OutputResource – The resource containing the converted document

The following conversion libraries will be used for text document conversion.

4.2 DOCFRAC

DOCFRAC (http://docfrac.sourceforge.net/)
Conversion Formats

- RTF to HTML
- RTF to TEXT
- HTML to RTF
- HTML to TEXT
- TEXT to RTF
- TEXT to HTML

Uses

- converting many documents at a time;
- active web pages; and
- converting output from Microsoft's Internet Explorer RTF control to HTML.

Platforms

- Windows;
- Linux command line; and
- programming kit (ActiveX and DLL).

DocFrac is free. It is released under the LGPL.

4.3 GNU Ghostscript

GNU Ghostscript (http://www.cs.wisc.edu/~ghost/)
Ghostscript is the name of a set of software that provides:

- An interpreter for the PostScript (TM) language and the Adobe Portable Document Format, and
- A set of C procedures (the Ghostscript library) that implement the graphics and filtering (data compression / decompression / conversion) capabilities that appear as primitive operations in the PostScript language and in PDF.

Versions entitled "GNU Ghostscript" are distributed with the GNU General Public License.

4.4 XPDF

XPDF (http://www.foolabs.com/xpdf/)
Xpdf is an open source viewer for Portable Document Format (PDF) files. The Xpdf project also includes a PDF text extractor, PDF-to-PostScript converter, and various other utilities.
Xpdf runs under the X Window System on UNIX, VMS, and OS/2. The non-X components (pdftops, pdftotext, etc.) also run on Win32 systems and should run on pretty much any system with a decent C++ compiler.
Xpdf is designed to be small and efficient. It can use Type 1, TrueType, or standard X fonts.
Xpdf is licensed under the GNU General Public License (GPL), version 2.

4.5 HTMLDOC

HTMLDOC (http://www.easysw.com/htmldoc/)
HTMLDOC converts HTML source files into indexed HTML, PostScript, or Portable Document Format (PDF) files that can be viewed online or printed.
The program is free software and is distributed under GPL.

Our plug-in is available in source code on the AXMEDIS CVS repository.

5 Adaptation Tools and Algorithms for Video (FHGIGD)

5.1 FFMPEG

Module/Tool Profile Tools and algorithms for Video Adaptation		
Responsible Name	Schmucker	
Responsible Partner	FHGIGD	
Status (proposed/approved)	Proposed	
Implemented/not implemented	Implemented	
Status of the implementation	100%	
Executable or Library/module (Support)	Library	
Single Thread or Multithread	Single	
Language of Development	C++	
Platforms supported	Currently implemented for Win32 (see above)	
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/newrepos/Framework/include/adaptation/video/ https://cvs.axmedis.org/newrepos/Framework/lib/adaptation/video/ https://cvs.axmedis.org/newrepos/Framework/project/adaptation/video/ https://cvs.axmedis.org/newrepos/Framework/source/adaptation/video/	
Reference to the AXFW location of the demonstrator executable tool for internal download	https://cvs.axmedis.org/newrepos/Framework/bin/adaptation/video/	
Reference to the AXFW location of the demonstrator executable tool for public download	http://www.axmedis.org	
Address for accessing to WebServices if any, add accession information (user aNd Passwd) if any	-	
Test cases (present/absent)	Absent	
Test cases location	-	
Usage of the AXMEDIS configuration manager (yes/no)	No	
Usage of the AXMEDIS Error Manager (yes/no)	No	
Major Problems not solved	Not all file formats can be handled with this library.	
Major pending requirements	--	
Interfaces API with other tools, named as	Name of the communicating tools References to other major components needed	Communication model and format (protected or not, etc.)
Formats Used	Shared with	format name or reference to a section
Protocol Used	Shared with	Protocol name or reference to a section
Used Database name		

User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
FFMPEG	Version 4.08	LGPL

5.1.1 General Description of the Module

The adaptation tools and algorithms for video provide the basic functions that are needed to convert video files. The module, which is initially provided, is based on the FFMPEG library (http://ffmpeg.sourceforge.net/index.php).

FFmpeg is “a set of free computer programs that can record, convert and stream digital audio and video. It includes libavcodec, a leading audio/video codec library. FFmpeg is developed under Linux, but it can compile under most operating systems, including Windows ... Notable is that most FFmpeg developers are part of either the MPlayer, xine or VideoLAN project as well.” (see http://en.wikipedia.org/wiki/FFmpeg)

The integrated plug-in, which is based on the FFMPEG functionality licensed under LGPG, doesn’t need to be configured. It was developed to be used in the AXMEDIS applications that allow the usage of AXMEDIS plug-ins. These include the AXEditor and the AXRuleEditor.

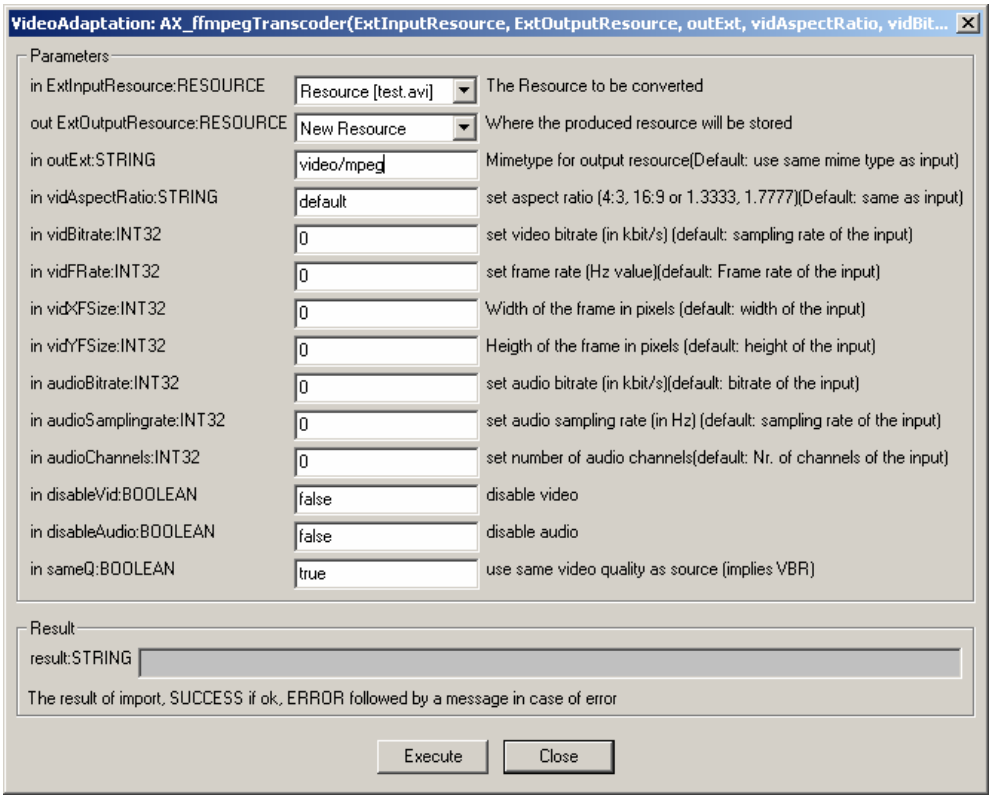
The implemented version was tested on MS Windows platform. Due to the platform independence of FFMPEG a conversion to other platforms is possible without spending too much effort on the core video adaptation functionality.

5.1.2 Module Design in terms of Classes

The main work consists of the plug-in integration. The integration is done in the class `VideoAdaptation`. This library is a wrapper class that calls the corresponding functionalities of the FFMPEG library.

5.1.3 User interface description

Usage of the developed plug-in depends on the AXMEDIS program that utilizes the available functionality. As shown in the next figure – which is an example when the plug-in is used in the AXEditor – several parameters can be set.



5.1.4 Technical and Installation information

The plug-in is designed to run on Microsoft Windows 2000/NT/XP platforms (it has been tested on Microsoft Windows XP professional and Microsoft Windows 2000). At least a Pentium III processor and 256MB RAM are needed. 6MB of disk space is needed.

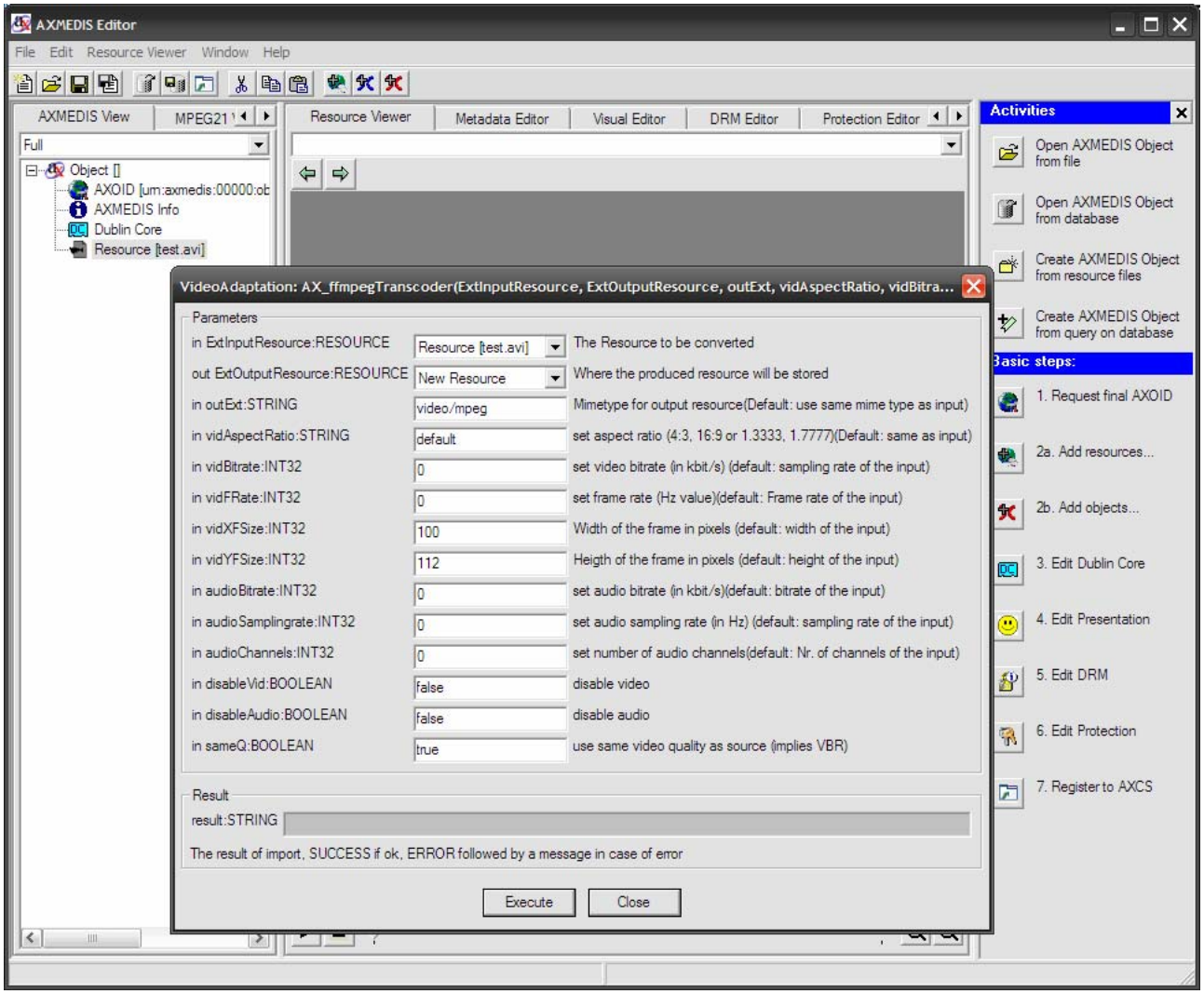
References to other major components needed	Dynamic Link Libraries (DLLs) of the FFMPEG library (avcodec-51_AXFP.dll, avformat-50_AXFP.dll, avutil-49_AXFP and ffmpeglib-50_AXFP)
Problems not solved	some AV codes are not supported (see: http://ffmpeg.mplayerhq.hu/ffmpeg-doc.html#SEC21)
Configuration and execution context	

5.1.5 Draft User Manual and Example of Usage

The video adaptation plug-in can be applied to the MIME type avi, mpeg, and video. This example shows the usage of the developed plug-in with the AXEditor.

Before using the video adaptation an AXMEDIS object containing a video resource has to be opened. Alternatively, a new AXMEDIS object can be created and a video resource has to be added.

For the video resource the “Content Processing Plug-in …” command has to be selected. Within this dialog, the VideoAdaptation plug-in has to be chosen.



The only functionality available is the video adaptation. A video resource is transcoded to the format as specified by the parameters that have been entered in the dialogue.

More detailed information is available in the plug-in description.

5.1.6 Errors reported and that may occur

Error code	Description and rationales
`ERROR: Unknown output Mime Type!`	MIME type of the output resource is unknown. As a consequence, the output resource cannot be generated.
`ERROR: Error resolving vidAspectRatio Parameter!`	Unknown aspect ratio was chosen
`ERROR: Error resolving Parameter!`	Unknown value for vidBitrate
`ERROR: Error resolving vidFRate Parameter!`	Unknown value for vidFRate
`ERROR: Error resolving vidXFSize Parameter!`	Unknown value for vidXFSize
`ERROR: Error resolving vidYFSize Parameter!`	Unknown value for vidYFSize
`ERROR: Error resolving audioBitrate Parameter!`	Unknown value for audioBitrate
`ERROR: Error resolving audioSamplingrate Parameter!`	Unknown value for audioSamplingrate
`ERROR: Error resolving audioChannels Parameter!`	Unknown value for audioChannels

`ERROR: Client provided a corrupted Input Stream!`	Input stream is not readable
`ERROR: Could not open decode input Mimetype:`	Input stream is not supported

5.1.7 Formal description of the algorithm

VideoAdaptation	
Method	VideoAdaptation
Description	This module allows the conversion of visual content/video content.
Input parameters	

ExtInputResource	The Resource to be converted
outExt	Mimetype for output resource (Default: use same mime type as input)
vidAspectRatio	set aspect ratio (4:3, 16:9 or 1.3333, 1.7777) (Default: same as input)
vidBitrate	set video bitrate (in kbit/s) (default: sampling rate of the input)
vidFRate	set frame rate (Hz value) (default: Frame rate of the input)
vidXFSize	Width of the frame in pixels (default: width of the input)
vidYFSize	Height of the frame in pixels (default: height of the input)
audioBitrate	set audio bitrate (in kbit/s) (default: bitrate of the input)
audioSamplingrate	set audio sampling rate (in Hz) (default: sampling rate of the input)
audioChannels	set number of audio channels (default: Nr. of channels of the input)
disableVid	disable video
disableAudio	disable audio
sameQ	use same video quality as source (implies VBR)

Output parameters	
ExtOutputResource	Where the produced resource will be stored

6 Adaptation Tools and Algorithms for Images (DSI)

6.1 ImageMagick

Module/Tool Profile Tools and Algorithms for Images Adaptation		
Responsible Name	Ivan Bruno	
Responsible Partner	DSI	
Status (proposed/approved)	Approved	
Implemented/not implemented	Implemented	
Status of the implementation	Complete	
Executable or Library/module (Support)	Library	
Single Thread or Multithread	Single	
Language of Development	C++	
Platforms supported	Windows and probably also LINUX and MAC	
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/newrepos/Framework/source/adaptation/image/ https://cvs.axmedis.org/newrepos/Framework/include/adaptation/image/ https://cvs.axmedis.org/newrepos/Framework/project/adaptation/image/win32/	
Reference to the AXFW location of the demonstrator executable tool for internal download		
Reference to the AXFW location of the demonstrator executable tool for public download		
Address for accessing to WebServices if any, add accession information (user aNd Passwd) if any		
Test cases (present/absent)		
Test cases location	http://////////////////	
Usage of the AXMEDIS configuration manager (yes/no)	No	
Usage of the AXMEDIS Error Manager (yes/no)	No	
Major Problems not solved	-- --	
Major pending requirements	-- --	
Interfaces API with other tools, named as	Name of the communicating tools References to other major components needed	Communication model and format (protected or not, etc.)
Formats Used	Shared with	format name or reference to a section

Protocol Used	Shared with	Protocol name or reference to a section
Used Database name		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
Image Magick	Version 6.1.9.4	Similar to LGPL, ImageMagick is available for free, may be used to support both open and proprietary applications, and may be redistributed without fee.

6.1.1 General Description of the Module

This module gives the possibility to use algorithms and tools for adaptation of images. The main adaptation functions needed by the AXMEDIS Framework could be summarised in:

- Scaling
- Resolution improvement/reduction
- Colour to Greyscale conversions
- Format transcoding
- Composition with other images
- Adding widgets and graphic motifs
- Text drawing
- Image decomposition

These functions are implemented by defining specific algorithms or using graphic libraries. An example of library is given by the *ImageMagick Library*.

6.1.2 Module Design in terms of Classes

No relevant class diagram.

6.1.3 User interface description

The Image adaptation functionalities are to be used as plug-ins through the AXCP interface. The plug-in simply consists of a DLL and an XML file describing the functionalities of the DLL. Both the DLL and the XML description should be installed in the plug-in directory of the AXCP compliant tool using the plug-in.

References to other major components needed	No other components are needed
Problems not solved	NONE
Configuration and execution context	

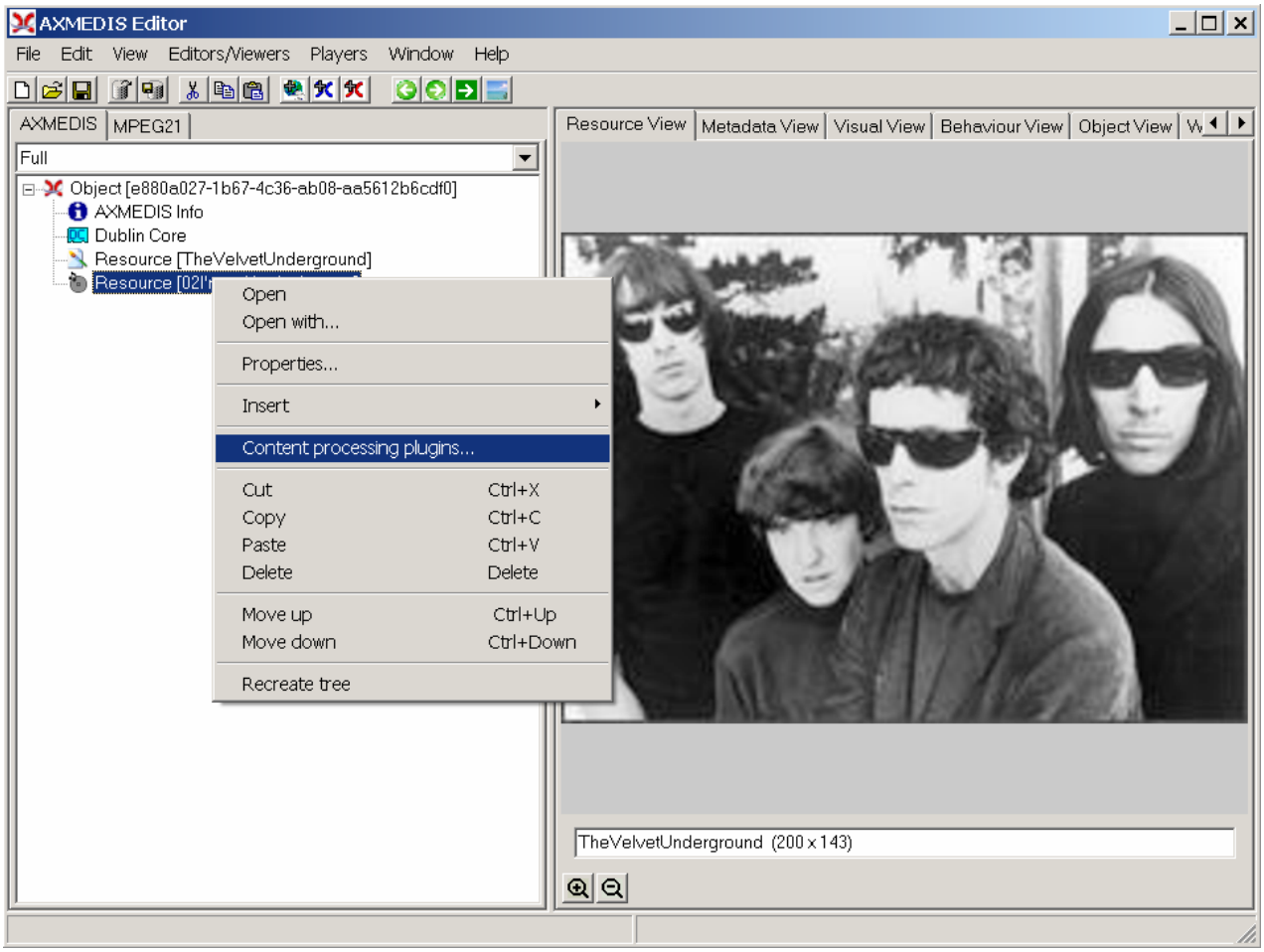
6.1.4 Draft User Manual

The Image Adaptation functions can be used to edit, adapt, transcode, etc an image file into a different format. For the Javascript usage please see the AXCP Area Specification Document.

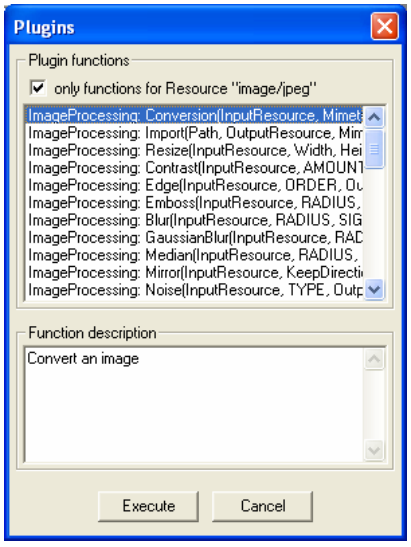
6.1.5 Examples of usage

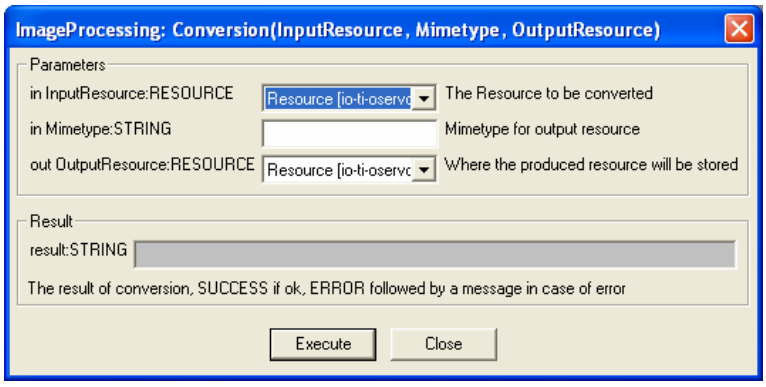
Here's an example on how to use the image adaptation functions as a plug-in with for the AXMEDIS editor.

The plug-in must be applied on an image resource of an AXMEDIS object. The adaptation plug-in is called by right-clicking on the interesting resource and selecting the 'Plugin…' command:



A window a) showing the functionalities available for the kind of resource selected appears:

a) 

b) 

The first image adaptation function available is the Conversion function which is selected by clicking on **ImageProcessing: Conversion**. A new window b) appears showing the interface to the Conversion function. In the example of the following figure, the conversion function is used to convert the image into another in different format:

- The first parameter is the selected image resource
- The Mimetype parameter define the output format
- The output resource allows specifying if the input resource has to be replaced with the converted image or a new one has to be generated.

6.1.6 Formal description of algorithm Conversion

Description: Convert an image in different formats.

Signature:

string Conversion (*AxResource* InputResource, *string* Mimetype, *AxResource* OutputResource)

Parameter List:

Name: InputResource
Description: The Resource to be converted
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:

Name: Mimetype
Description: Mimetype for output resource
Paramater Type *string*
Default Value:
Constraints:

Name: OutputResource
Description: Where the produced resource will be stored
Paramater Type *AxResource*
Default Value:

Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.7 Formal description of algorithm Import

Description: Import an image

Signature:
string Import (*string* Path, *AxResource* OutputResource, *string* MimeType)
Parameter List:
Name: Path
Description: Path to the image
Paramater Type *string*
Default Value:
Constraints:
Name: OutputResource
Description: Where the imported resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: MimeType
Description: Mimetype for testing
Paramater Type *string*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.8 Formal description of algorithm Resize

Description: Resizes an image

Signature:
string Resize (*AxResource* InputResource, INT32 Width, INT32 Height, BOOLEAN KeepAspectRatio, *AxResource* OutputResource)

Parameter List:
Name: InputResource
Description: The Resource to be resized
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: Width
Description: The new image width
Paramater Type INT32

Default Value:
Constraints:
Name: Height
Description: The new image height
Paramater Type INT32
Default Value:
Constraints:
Name: KeepAspectRatio
Description: Indicates to preserve image aspect ratio or not
Paramater Type BOOLEAN
Default Value:
Constraints:
Name: OutputResource
Description: Where the resized resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.9 Formal description of algorithm Contrast

Description: Change image contrast

Signature:
string Contrast (*AxResource* InputResource, INT32 AMOUNT, *AxResource* OutputResource)

Parameter List:
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: AMOUNT
Description: The contrast amount
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.10 Formal description of algorithm Edge

Description: Edge image (hilight edges in image). The radius is the radius of the pixel neighborhood.. Specify a radius of zero for automatic radius selection.

Signature:
string Edge (*AxResource* InputResource, INT32 ORDER, *AxResource* OutputResource)

Parameter List:
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: ORDER
Description: The Order Edge
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.11 Formal description of algorithm Emboss

Description: Emboss image (hilight edges with 3D effect). The radius_ parameter specifies the radius of the Gaussian, in pixels, not counting the center pixel. The sigma_ parameter specifies the standard deviation of the Laplacian, in pixels.

Signature:
string Emboss (*AxResource* InputResource, INT32 RADIUS, INT32 SIGMA, *AxResource* OutputResource)

Parameter List:
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: RADIUS
Description: The Radius Emboss
Paramater Type INT32
Default Value:
Constraints:

Name: SIGMA
Description: The sigma Emboss
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.12 Formal description of algorithm Blur

Description: Blur image.

Signature:
string Blur (*AxResource* InputResource, INT32 RADIUS, INT32 SIGMA, *AxResource* OutputResource)

Parameter List:
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: RADIUS
Description: The Radius Blur
Paramater Type INT32
Default Value:
Constraints:
Name: SIGMA
Description: The sigma Blur
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.13 Formal description of algorithm GaussianBlur

Description: GaussianBlur the image

Signature:
string GaussianBlur (*AxResource* InputResource, INT32 RADIUS, INT32 SIGMA, *AxResource* OutputResource)

Parameter List:
Name: InputResource
Description: Gaussian blur image. The number of neighbor pixels to be included in the convolution mask is specified by 'width_'. For example, a width of one gives a (standard) 3x3 convolution mask. The standard deviation of the gaussian bell curve is specified by 'sigma'.
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: RADIUS
Description: The Radius GaussianBlur
Paramater Type INT32
Default Value:
Constraints:
Name: SIGMA
Description: The sigma GaussianBlur
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.14 Formal description of algorithm Median

Description: Median the image

Signature:
string Median (*AxResource* InputResource, INT32 RADIUS, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: RADIUS
Description: The Radius Median
Paramater Type INT32
Default Value:

Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.15 Formal description of algorithm Mirror

Description: Mirror the image

Signature:
string Mirror (*AxResource* InputResource, BOOLEAN KeepDirection, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: KeepDirection
Description: The KeepDirection Mirror
Paramater Type BOOLEAN
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.16 Formal description of algorithm Noise

Description: Noise the image

Signature:
string Noise (*AxResource* InputResource, INT32 TYPE, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:

Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: TYPE
Description: The Type Noise
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.17 Formal description of algorithm Despeckle

Description: Despeckle image (reduce speckle noise)

Signature:
string Despeckle (*AxResource* InputResource, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.18 Formal description of algorithm Equalize

Description: Equalize image (histogram equalization)

Signature:
string Equalize (*AxResource* InputResource, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*

Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.19 Formal description of algorithm Enhance

Description: Enhance image (minimize noise)

Signature:
string Enhance (*AxResource* InputResource, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.20 Formal description of algorithm ExtractChannel

Description: Extract Channel of the image

Signature:
string ExtractChannel (*AxResource* InputResource, INT32 CHANNEL, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png

Ranges:
Name: CHANNEL
Description: The Channel ExtractChannel
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.21 Formal description of algorithm Grayscale

Description: Grayscale the image

Signature:
string Grayscale (*AxResource* InputResource, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.22 Formal description of algorithm Magnify

Description: Magnify image by integral size

Signature:
string Magnify (*AxResource* InputResource, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:

Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.23 Formal description of algorithm Minify

Description: Reduce image by integral size

Signature:
string Minify (*AxResource* InputResource, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.24 Formal description of algorithm Modulate

Description: Modulate percent hue, saturation, and brightness of an image.

Signature:
string Modulate (*AxResource* InputResource, INT32 BRIGHTNESS, INT32 SATURATION, INT32 HUE, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:

Name: BRIGHTNESS
Description: Brightness modulate
Paramater Type INT32
Default Value:
Constraints:
Name: SATURATION
Description: Saturation modulate
Paramater Type INT32
Default Value:
Constraints:
Name: HUE
Description: Hue modulate
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.25 Formal description of algorithm Monochrome

Description: Monochrome the image

Signature:
string Monochrome (*AxResource* InputResource, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.26 Formal description of algorithm Negate

Description: Negate colors in image. Replace every pixel with its complementary color (white becomes black, yellow becomes blue, etc.). Set grayscale to only negate grayscale values in image.

Signature:
string Negate (*AxResource* InputResource, BOOLEAN GRAYSCALE, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: GRAYSCALE
Description: Where the manipulated resource will be stored
Paramater Type BOOLEAN
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.27 Formal description of algorithm Normalize

Description: Normalize image (increase contrast by normalizing the pixel values to span the full range of color values)

Signature:
string Normalize (*AxResource* InputResource, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.28 Formal description of algorithm OilPaint

Description: Oilpaint image (image looks like oil painting)

Signature:
string OilPaint (*AxResource* InputResource, INT32 RADIUS, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: RADIUS
Description: the radius OilPaint
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.29 Formal description of algorithm Quality

Description: JPEG/MIFF/PNG compression level (default 75).

Signature:
string Quality (*AxResource* InputResource, INT32 LEVEL, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: LEVEL
Description: the quality of the compress level
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:

Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.30 Formal description of algorithm Quantize

Description: Preferred number of colors in the image. The actual number of colors in the image may be less than your request, but never more. Images with less unique colors than specified with this option will have any duplicate or unused colors removed.

Signature:
string Quantize (*AxResource* InputResource, INT32 NCOLORS, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: NCOLORS
Description: the number of color
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.31 Formal description of algorithm Raise

Description: Raise image (lighten or darken the edges of an image to give a 3-D raised or lowered effect)

Signature:
string Raise (*AxResource* InputResource, INT32 WIDTH, INT32 HEIGHT, INT32 XOFFSET, INT32 YOFFSET, BOOLEAN RISED, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:

Name: WIDTH
Description: The width is parts of the geometry specification are measured in pixels
Paramater Type INT32
Default Value:
Constraints:
Name: HEIGHT
Description: The height is parts of the geometry specification are measured in pixels
Paramater Type INT32
Default Value:
Constraints:
Name: XOFFSET
Description: The left edge of the object is to be placed xoffset pixels in from the left edge of the image.
Paramater Type INT32
Default Value:
Constraints:
Name: YOFFSET
Description: The top edge of the object is to be yoffset pixels below the top edge of the image.
Paramater Type INT32
Default Value:
Constraints:
Name: RISED
Description: raisedFlag
Paramater Type BOOLEAN
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.32 Formal description of algorithm ReduceNoise

Description: Reduce noise in image using a noise peak elimination filter.

Signature:
string ReduceNoise (*AxResource* InputResource, INT32 ORDER, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: ORDER
Description: order
Paramater Type INT32

Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.33 Formal description of algorithm Replace

Description: Replace the image

Signature:
string Replace (*AxResource* InputResource, INT32 R1, INT32 G1, INT32 B1, INT32 R2, INT32 G2, INT32 B2, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: R1
Description: r1
Paramater Type INT32
Default Value:
Constraints:
Name: G1
Description: g1
Paramater Type INT32
Default Value:
Constraints:
Name: B1
Description: b1
Paramater Type INT32
Default Value:
Constraints:
Name: R2
Description: r2
Paramater Type INT32
Default Value:
Constraints:
Name: G2
Description: g2
Paramater Type INT32
Default Value:
Constraints:
Name: B2

Description: b2
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.34 Formal description of algorithm FloodFill

Description: Flood-fill texture across pixels that match the color of the target pixel and are neighbors of the target pixel. Uses current fuzz setting when determining color match.

Signature:
string FloodFill (*AxResource* InputResource, INT32 X, INT32 Y, INT32 B, INT32 R, INT32 G, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: X
Description: x
Paramater Type INT32
Default Value:
Constraints:
Name: Y
Description: y
Paramater Type INT32
Default Value:
Constraints:
Name: B
Description: b
Paramater Type INT32
Default Value:
Constraints:
Name: R
Description: r
Paramater Type INT32
Default Value:
Constraints:
Name: G
Description: g
Paramater Type INT32

Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.35 Formal description of algorithm Roll

Description: Roll image (rolls image vertically and horizontally) by specified number of columns and rows)

Signature:
string Roll (*AxResource* InputResource, INT32 X, INT32 Y, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: X
Description: x
Paramater Type INT32
Default Value:
Constraints:
Name: Y
Description: y
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.36 Formal description of algorithm Rotate

Description: Rotate image counter-clockwise by specified number of degrees.

Signature:
string Rotate (*AxResource* InputResource, INT32 ANGLE, *AxResource* OutputResource)

Parameter List

Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: ANGLE
Description: Number of the degrees
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.37 Formal description of algorithm Scale

Description: Resize image by using simple ratio algorithm

Signature:
string Scale (*AxResource* InputResource, INT32 WIDTH, INT32 HEIGHT, INT32 MODE, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: WIDTH
Description: Width
Paramater Type INT32
Default Value:
Constraints:
Name: HEIGHT
Description: Height
Paramater Type INT32
Default Value:
Constraints:
Name: MODE
Description: Mode
Paramater Type INT32
Default Value:
Constraints:

Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.38 Formal description of algorithm Shear

Description: Shear image (create parallelogram by sliding image by X or Y axis). Shearing slides one edge of an image along the X or Y axis, creating a parallelogram. An X direction shear slides an edge along the X axis, while a Y direction shear slides an edge along the Y axis. The amount of the shear is controlled by a shear angle. For X direction shears, x degrees is measured relative to the Y axis, and similarly, for Y direction shears y degrees is measured relative to the X axis. Empty triangles left over from shearing the image are filled with the color defined as borderColor.

Signature:
string Shear (*AxResource* InputResource, INT32 XSHEAR, INT32 Yshear, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: XSHEAR
Description: XSHEAR
Paramater Type INT32
Default Value:
Constraints:
Name: Yshear
Description: Yshear
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.39 Formal description of algorithm Shade

Description: Shade image using distant light source. Specify azimuth_ and elevation_ as the position of the light source. By default, the shading results as a grayscale image. Set colorShading_ to true to shade the red, green, and blue components of the image.

Signature:
string Shade (*AxResource* InputResource, INT32 AZIMUTH, INT32 ELEVATION, BOOLEAN COLOR, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: AZIMUTH
Description: AZIMUTH
Paramater Type INT32
Default Value:
Constraints:
Name: ELEVATION
Description: ELEVATION
Paramater Type INT32
Default Value:
Constraints:
Name: COLOR
Description: COLOR
Paramater Type BOOLEAN
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.40 Formal description of algorithm Spread

Description: Spread pixels randomly within image by specified amount.

Signature:
string Spread (*AxResource* InputResource, INT32 AMOUNT, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:

Name: AMOUNT
Description: AMOUNT
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.41 Formal description of algorithm SetOpacity

Description: Set the opacity of the image.

Signature:
string SetOpacity (*AxResource* InputResource, INT32 LEVEL, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: LEVEL
Description: LEVEL
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.42 Formal description of algorithm SubImage

Description: SubImage image.

Signature:
string SubImage (*AxResource* InputResource, INT32 X, INT32 Y, INT32 WIDTH, INT32 HEIGHT, *AxResource* OutputResource)

Parameter List
Name: InputResource

Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: X
Description: x coordinate of the top-level corner of the rectangle
Paramater Type INT32
Default Value:
Constraints:
Name: Y
Description: y coordinate of the top-level corner of the rectangle
Paramater Type INT32
Default Value:
Constraints:
Name: WIDTH
Description: Width member
Paramater Type INT32
Default Value:
Constraints:
Name: HEIGHT
Description: Height member
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.43 Formal description of algorithm GetInfo

Description: Return the size of the image.

Signature:
string GetInfo (*AxResource* InputResource, INT32 WIDTH, INT32 HEIGHT)

Parameter List
Name: InputResource
Description: The Resource under analisys
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: WIDTH
Description: The width of the Image

Paramater Type INT32
Default Value:
Constraints:
Name: HEIGHT
Description: The height of the Image
Paramater Type INT32
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.44 Formal description of algorithm SetMaskColour

Description: Set the color

Signature:
string SetMaskColour (*AxResource* InputResource, INT32 R, INT32 G, INT32 B, *AxResource* OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: R
Description: Red
Paramater Type INT32
Default Value:
Constraints:
Name: G
Description: Green
Paramater Type INT32
Default Value:
Constraints:
Name: B
Description: Blue
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.45 Formal description of algorithm Paste

Description: Paste image

Signature:
string Paste (*AxResource* InputResource1, *AxResource* InputResource2, INT32 X, INT32 Y, INT32 COMPOSE, *AxResource* OutputResource)

Parameter List
Name: InputResource1
Description: The Resource to be manipulated
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: InputResource2
Description: The Resource paste
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: X
Description: X
Paramater Type INT32
Default Value:
Constraints:
Name: Y
Description: Y
Paramater Type INT32
Default Value:
Constraints:
Name: COMPOSE
Description: Compose
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

6.1.46 Formal description of algorithm Test

Description: Test an image

Signature:

AxResource Test (*AxResource* InputResource, AXOM Axom)

Parameter List

Name: InputResource
Description: The Resource to be tested
Paramater Type *AxResource*
Default Value:
Constraints:

Name: Axom
Description: The object
Paramater Type AXOM
Default Value:
Constraints:

Result: Result
Result type: *AxResource*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

7 Adaptation Tools and Algorithms for Audio (EPFL)

7.1 General

Module/Tool Profile Tools and Algorithms for Audio Adaptation		
Responsible Name	Mattavelli	
Responsible Partner	EPFL	
Status (proposed/approved)	Proposed	
Implemented/not implemented	Implemented	
Status of the implementation	Plug-In	
Executable or Library/module (Support)	Library	
Single Thread or Multithread	Single	
Language of Development	C++	
Platforms supported	Win32	
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/newrepos/Framework/src/adaptation/audio	
Reference to the AXFW location of the demonstrator executable tool for internal download	https://cvs.axmedis.org/newrepos/Framework/bin/adaptation/audio	
Reference to the AXFW location of the demonstrator executable tool for public download	--	
Address for accessing to WebServices if any, add accession information (user aNd Passwd) if any	--	
Test cases (present/absent)	Absent	
Test cases location	--	
Usage of the AXMEDIS configuration manager (yes/no)	No	
Usage of the AXMEDIS Error Manager (yes/no)	No	
Major Problems not solved	-- --	
Major pending requirements	-- --	
Interfaces API with other tools, named as	Name of the communicating tools References to other major components needed	Communication model and format (protected or not, etc.)
Formats Used	Shared with	format name or reference to a section

Protocol Used	Shared with	Protocol name or reference to a section
Used Database name		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
FFmpeg		LGPL
libsndfile		LGPL
SoundTouch		LGPL

7.1.1 Description of the Module

AXMEDIS media objects are to be distributed over heterogeneous networks and towards different kind of terminals. Moreover, the people who will ultimately consume and interact with the content may have different behaviours and preferences. Consequently, digital items should be adapted to fit their particular usage environment - this is the goal of AXMEDIS adaptation tools. More specifically, the adaptation tools should be able to modify content to fit:

- Terminal capabilities (codec, formats, input-output, etc supported by the terminal)
- Network characteristics (for example the minimum guaranteed bandwidth of a network)
- User characteristics (presentation preferences, auditory or visual impairment etc)
- Natural environment characteristics (for example the illumination characteristics that may affect the perceived display of visual information)

The seventh part of ISO/IEC 21000 (MPEG-21) specifies tools for the adaptation of Digital Items. More specifically, it proposes a set of normalized tools describing the usage environment of a digital item to command adaptation tools. According to what precedes and for the particular case of audio content, AXMEDIS Audio Adaptation tools should allow to adapt content according to the following MPEG-21 usage environment descriptors:

Terminal capabilities:

- **CodecCapabilities: specifies the decoding and encoding capabilities of a terminal.** Specifically, capabilities are defined by the format that a particular terminal is capable of encoding or decoding. Given the variety of different content representation formats that are available today, it is necessary

to be aware of the formats that a terminal is capable of. A terminal may be capable of both encoding and decoding and may also be capable of multiple formats.

- **AudioOutputCapabilities**: specifies audio output capabilities of the terminal. Describing the capabilities of an audio output indicates limitation that impacts the auditory presentation of information. This is achieved by specifying the sampling frequency and bits-per-sample, the frequency range of the output, the number of channels supported, as well as power and signal-to-noise ratio.

Network characteristics:

- **NetworkCapability**: specifies the static capabilities of a network, which includes attributes that describe the maximum capacity of a network and the minimum guaranteed bandwidth that a network can provide. Also specified are attributes that indicate if the network can provide in-sequence packet delivery and how the network deals with erroneous packets, i.e., does it forward, correct or discard them.

User characteristics:

- **AudioPresentationPreferences**: specifies the preferences of a User regarding the presentation or rendering or audio resources. Specifically, descriptions such as the preferred volume, frequency equalizer settings, and audible frequency ranges are specified. Such attributes may affect the way in which the delivered audio resource is encoded, e.g., allocating more bits to specific components in the given frequency range. Additionally, for limited capability devices that may not have equalization functionality, equalization may be performed prior to transmission given the designated preferences.
- **AuditoryImpairment:** describes the characteristics of a particular User's auditory deficiency. The description can be used by the audio resource adaptation engine to optimize the experience of audio contents for the User. The hearing threshold shift of a User is described. The description can be used to compensate the User's auditory impairment such as hearing loss during the adaptation. The tool is meant to be used in case of small hearing loss. For people with more than minimal impairments, gain changes to match the audiogram are normally not sufficient.

Natural environment characteristics:

- **AudioEnvironment:** describes the natural audio environment of a particular User in terms of the measured noise level and noise frequency spectrum. These descriptions can be used by audio resource adaptation engine to deliver the best experience of audio contents.

AXMEDIS Audio Adaptation Tools will provide the functionalities to transform audio content according to these usage environment characteristics. Required functionalities include notably:

- Format/codec transcoding
- Up/Downsampling
- Channels mixing
- Volume change
- Equalization

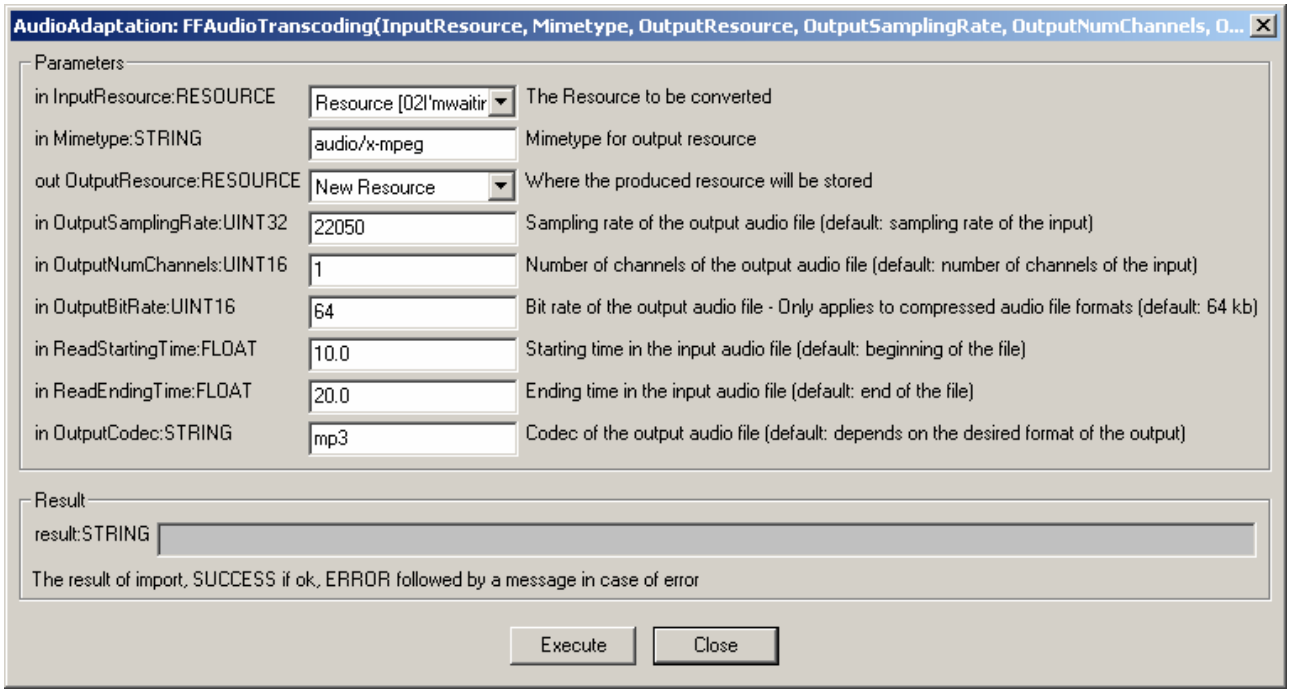
The following subsections describe the external library that will be used in the AXMEDIS framework to implement these functionalities.

7.1.2 User interface description

The audio adaptation functionalities are to be used as plug-ins through the AXCP interface. The interface of AXCP plug-ins maps exactly the formal description of the function and allows entering textually all parameters of the function. Moreover, it displays a brief description of the meaning of the parameters of the function to ease their use. The result of the adaptation is displayed as a textual message in the *Result* box of the interface.

The adaptation is launched by clicking the *Execute* button and the window can be closed with the *Close* button once the adaptation has been performed.

The following figure shows the user interface of the audio FFmpeg transcoding function. Please refer to sections 7.2.2 and 7.3.2 for the formal description of the audio transcoding function and to understand how the user interface reflects this formal description.



7.1.3 Technical and Installation information

The audio adaptation functionalities are to be used as plug-ins through the AXCP interface. The plug-in simply consists of a DLL and an XML file describing the functionalities of the DLL. Both the DLL and the XML description should be installed in the plug-in directory of the AXCP compliant tool using the plug-in.

References to other major components needed	The following DLLs need to be loaded for the correct execution of the audio adaptation tool. The simplest solution is to copy them into the directory of the AXCP compliant tool using the audio adaptation functionalities: • avcodec.dll • avformat.dll • libsndfilefile.dll • zlib1.dll
Problems not solved	• NONE
Configuration and execution context	--

7.1.4 Integration and compilation issues

The tool has been compiled and tested successfully on Win32 platform. It should be ported easily on Linux and MacOsX platforms though it has not been tested yet.

7.2 FFmpeg

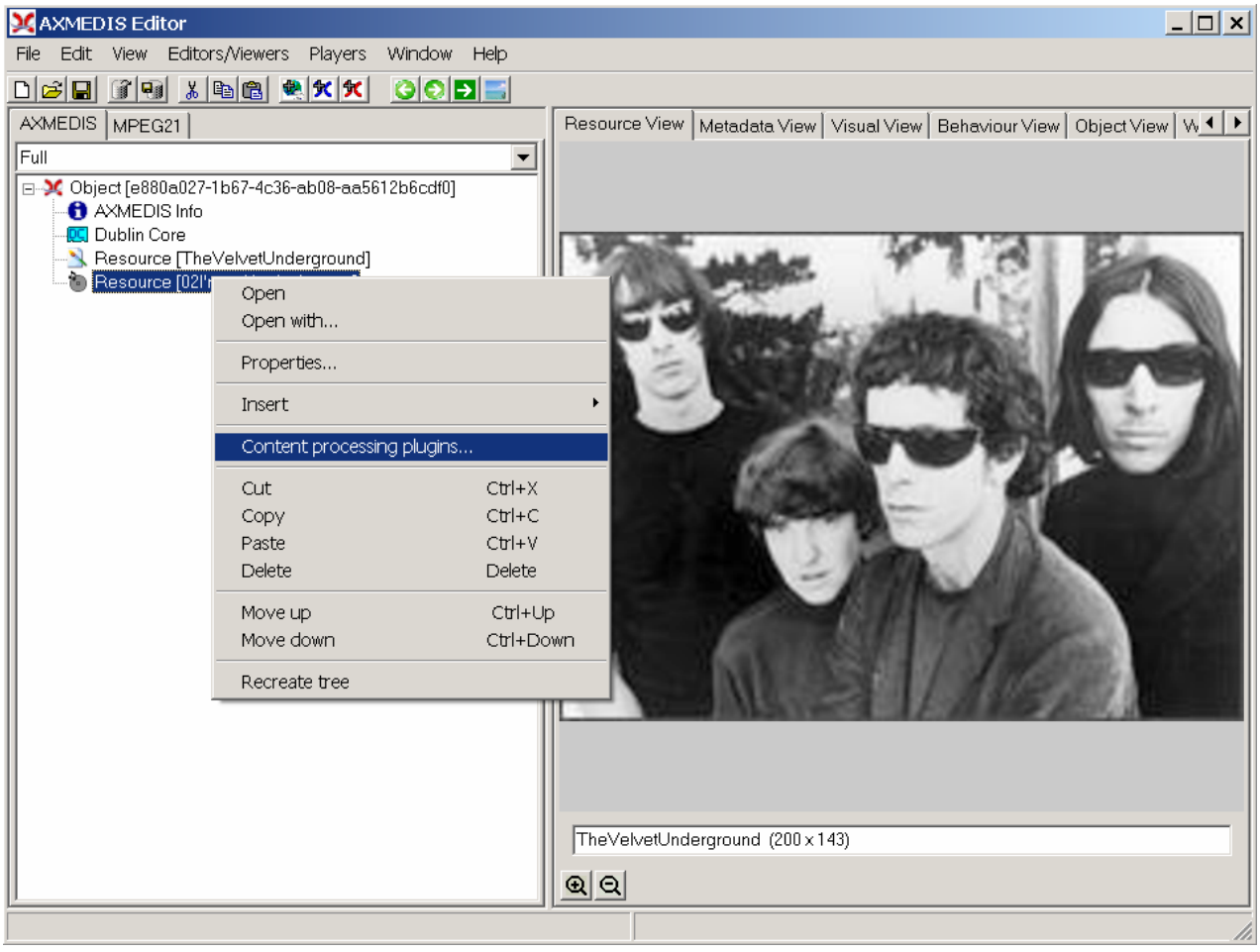
7.2.1 FFmpeg Audio Transcoding

The FFmpeg Audio Transcoding function can be used to convert an audio file into a different format and/or codec (please refer to section 4.4.10 for a complete description of formats and codecs supported in decoding and encoding). Apart from the bit rate reduction depending on the selected codec, one can further reduce the size of the resulting audio file by changing its sample rate and its number of audio channels. Moreover one can select only a specific portion of the input file to produce the resulting output file by specifying starting and ending points in the input file.

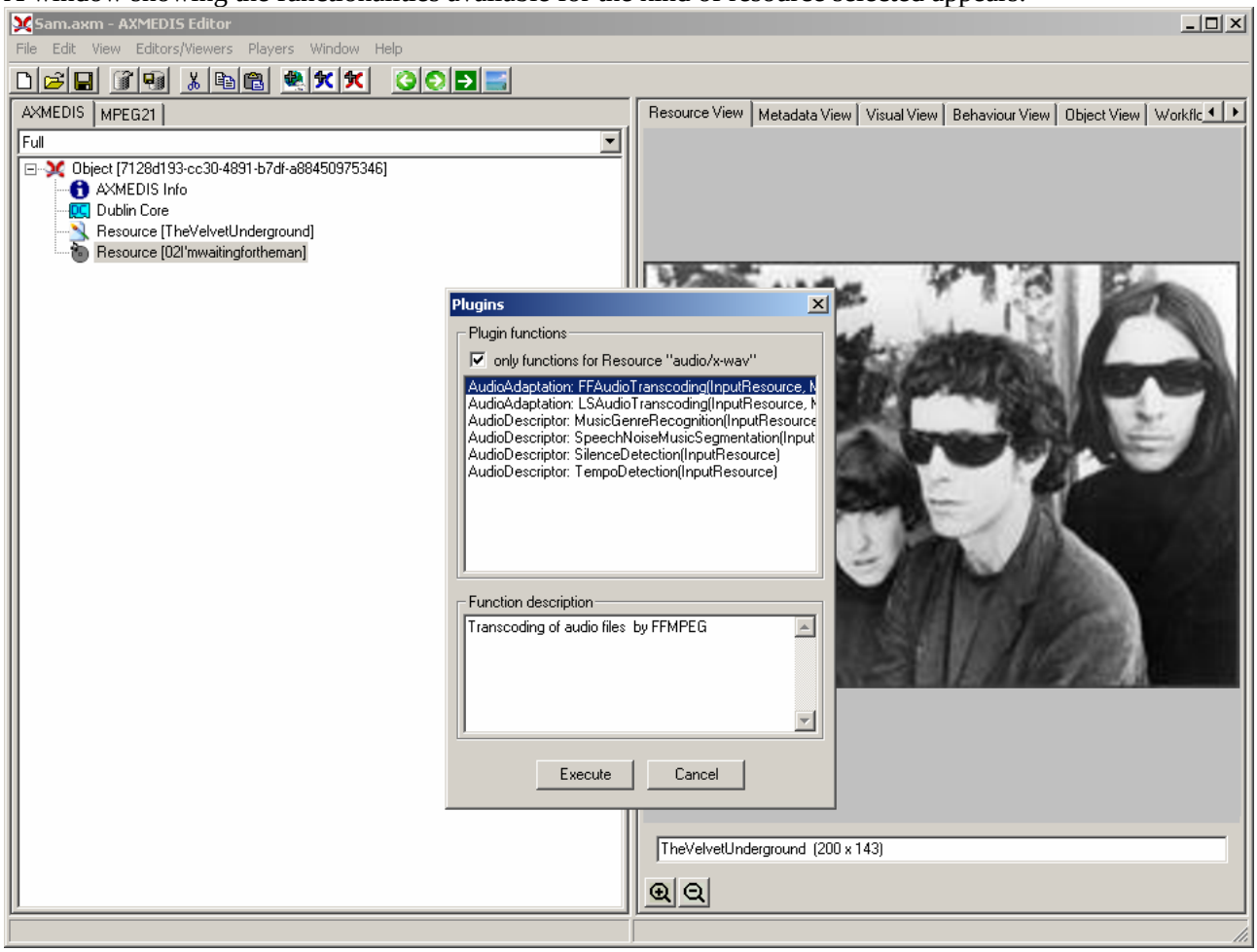
7.2.2 Draft User Manual

Here's an example on how to use the FFmpeg audio adaptation transcoding function as a plug-in with for the AXMEDIS editor.

The plug-in must be applied on an audio resource of an AXMEDIS object. The adaptation plug-in is called by right-clicking on the interesting resource and selecting the 'Plugin…' command:



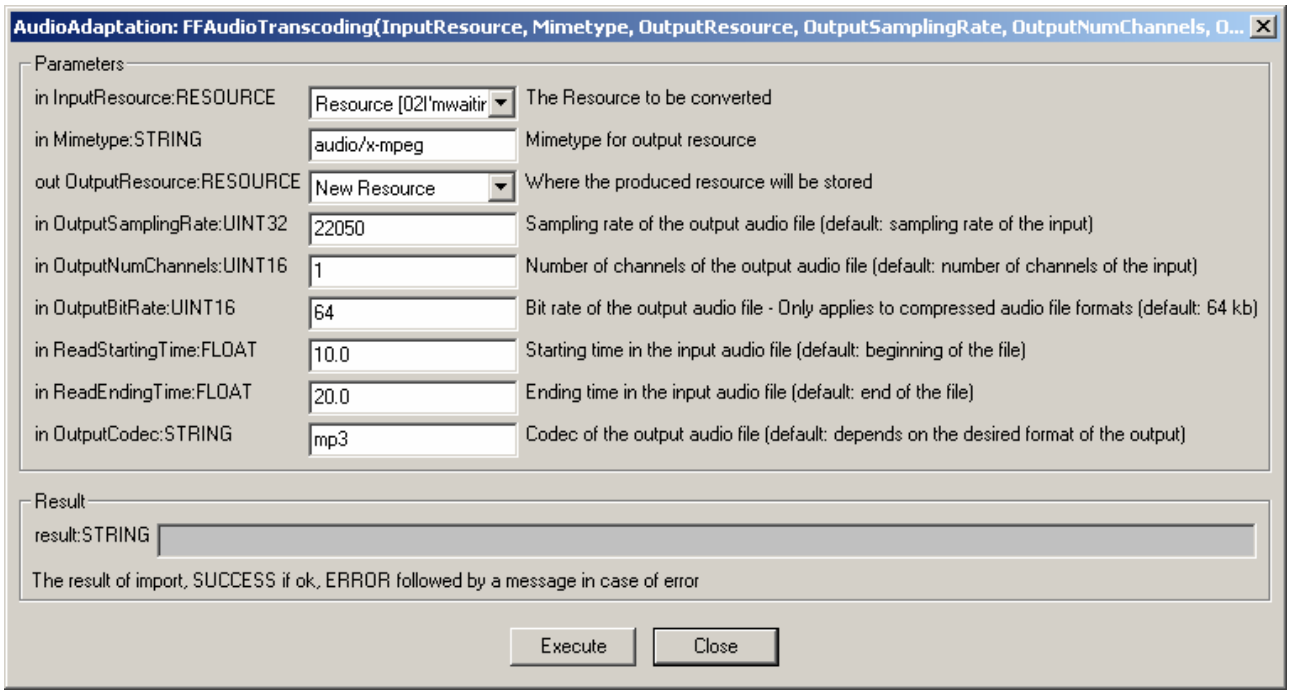
A window showing the functionalities available for the kind of resource selected appears:



The first audio adaptation function available is the FFmpeg transcoding function which is selected by clicking on **FFAudioAdaptation: FFAudioTranscoding**. A new window appears showing the interface to the audio transcoding function. In the example of the following figure, the transcoding function is used to create a 10 second snapshot with reduced bit rate of the input audio file:

- Mp3 compression is selected with a bit rate of 64 kB (which corresponds to a low quality)
- Further bit rate reduction is achieved by using a lower sampling rate for the output (22050 Hz) and mixing audio channels into a single mono channel
- Only a portion of 10 seconds of the input resource is selected (starting at time 10 seconds and ending at time 20 seconds)

A snapshot with reduced bit rate is particularly useful to allow a customer to pre-view an item before purchasing the corresponding high quality object.



7.2.3 Formal description of algorithm FFmpeg Transcoding

Description: encode an audio file in another format or another codec and change its sample rate and number of audio channels if needed.

Signature:

string Trancoding(*AxResource* InputResource, *string* MimeType, *AxResource* OutputResource, *UINT32* OutputSamplingRate, *UINT16*, OuputNumChannels, *UINT16* OutputBitRate, *float* ReadStartingTime, *float* ReadEndingTime, *string* OutputCodec)

Parameter List:

Name: InputResource
Description: the resource to be converted
Parameter Type: *AxResource*
Default Value:
Constraints:
Resource Type: audio
Resource Format: x-mpeg (.mp3), x.aiff (.aif, .aiff), x-wav (.wav), basic (.au, .snd), x-ms-wma (.wma), x-vorbis (.ogg), x-pn-realaudio (.ra, .ram)
Ranges:

Name: MimeType
Description: MimeType for the output resource
Parameter Type: *string*
Default Value:
Constraints:
Resource Type: audio
Resource Format: x-mpeg, x-aiff, x-wav, basic, x-vorbis, x-ac3
Ranges:

Name: OutputResource
Description: Where the output resource will be stored
Parameter Type: *AxResource*

Default Value:
Constraints:
Range:

Name: OutputSamplingRate
Description: The sampling rate of the output resource in Hertz
Parameter Type: *uint32*
Default Value: by default, the sampling rate of the input resource is used
Constraints:
Range:

Name: OutputNumChannels
Description: The number of channels of the audio resource after transcoding
Parameter Type: *uint16*
Default Value: by default, the number of channels of the input resource is used
Constraints:
Range:

Name: OutputBitRate
Description: The bit rate of the audio resource after transcoding in kilo-Bytes (this parameter is used when transcoding towards a compressed audio format such as MP3)
Parameter Type: *uint16*
Default Value: by default, the bit rate is set to 64 kB
Constraints:
Range:

Name: ReadStartingTime
Description: set the beginning of the output resource to *ReadStartingTime* seconds from the beginning of the input resource
Parameter Type: *float*
Default Value: by default, the read starting time is set to 0 seconds which means that the input resource is considered from the beginning
Constraints:
Range:

Name: ReadEndingTime
Description: set the end of the output resource at *ReadEndingTime* seconds from the beginning of the input resource
Parameter Type: *float*
Default Value: by default, the read ending time is set to the end of the input resource
Constraints:
Range:

Name: OutputCodec
Description: set the codec of the output resource; depending on the mime type selected for the output resource, only a certain subset of codec will be supported (the following table shows the possible codecs according to the possible mime types)
Parameter Type: *string*
Default Value: the default codec depend on the mime type selected for the output resource (the following table shows the default codec according to the possible mime types)
Constraints:
Range:

Result: Result
Result Type: string

Result Description: the result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

File Formats

For a list of codecs and formats supported by FFMPEG, please refer to section 34.1.
Mime type accepted

```
audio/x-wav
audio/x-ms-wma
audio/basic
audio/x-mpeg
audio/x-vorbis
audio/x-pn-realaudio
audio/x-ac3
audio/x-dv
audio/x-mace
audio/x-adpcm
audio/x-aac
audio/32KADPCM
audio/amr
video/x-mpeg
video/x-mpeg2
video/mp4
video/x-raw
video/x-h263
video/x-mjpeg
video/x-ms-wmv
video/x-ms-asf
video/x-flv
video/x-svq
video/x-dv
video/x-h264
video/x-indeo
video/x-vp3
video/x-ffv
video/x-vcr
video/x-msvideo
video/x-nut
application/x-pcm
application/vnd.rn-realmedia
```

7.3 Libsndfile

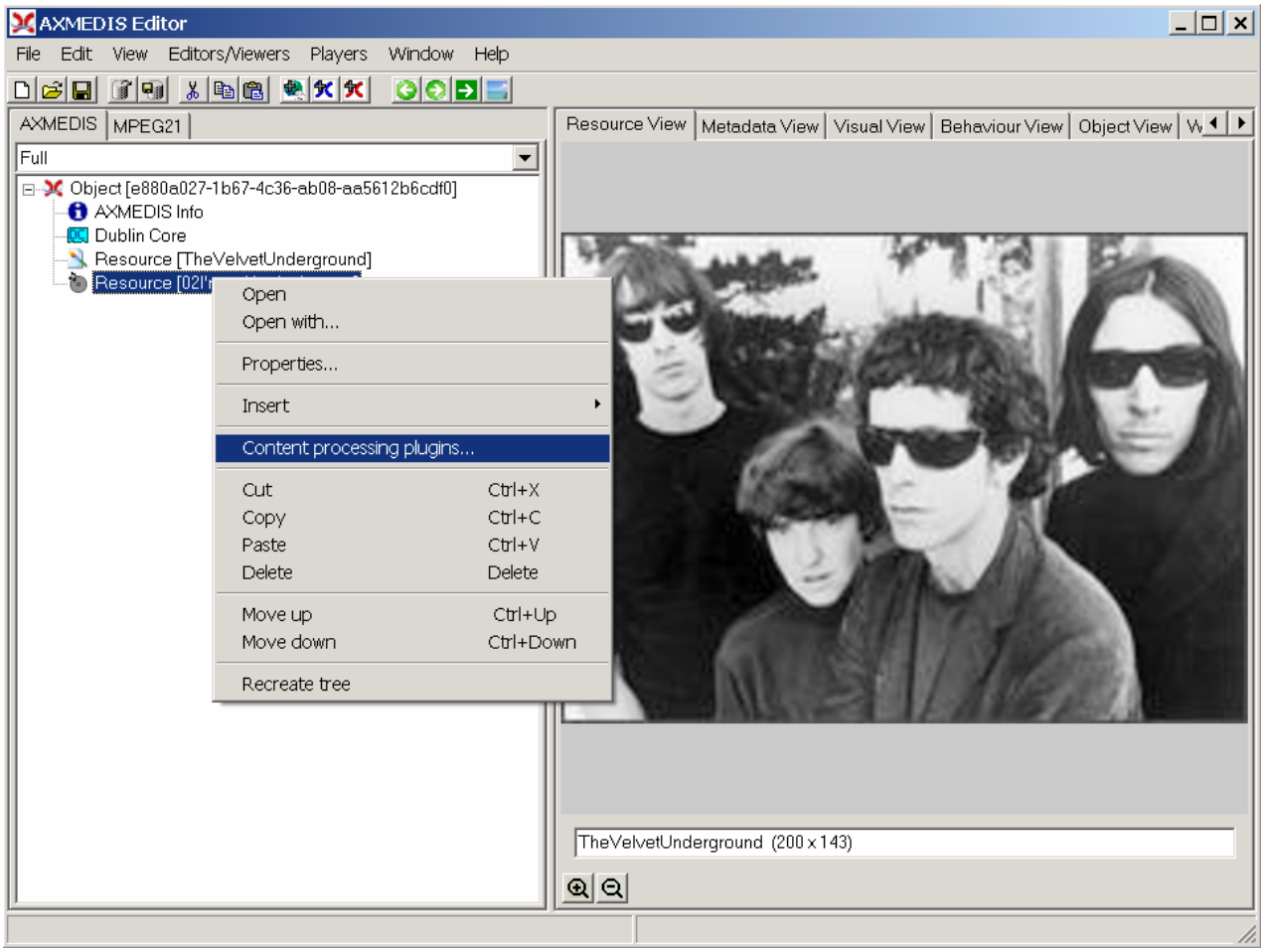
7.3.1 Libsndfile Audio Transcoding

The libsndfile Audio Transcoding function can be used to convert an audio file into a different format and/or codec (please refer to section 4.4.10 for a complete description of formats and codecs supported in decoding and encoding). Apart from the bit rate reduction depending on the selected codec, one can further reduce the size of the resulting audio file by changing its sample rate and its number of audio channels. Moreover one can select only a specific portion of the input file to produce the resulting output file by specifying starting and ending points in the input file.

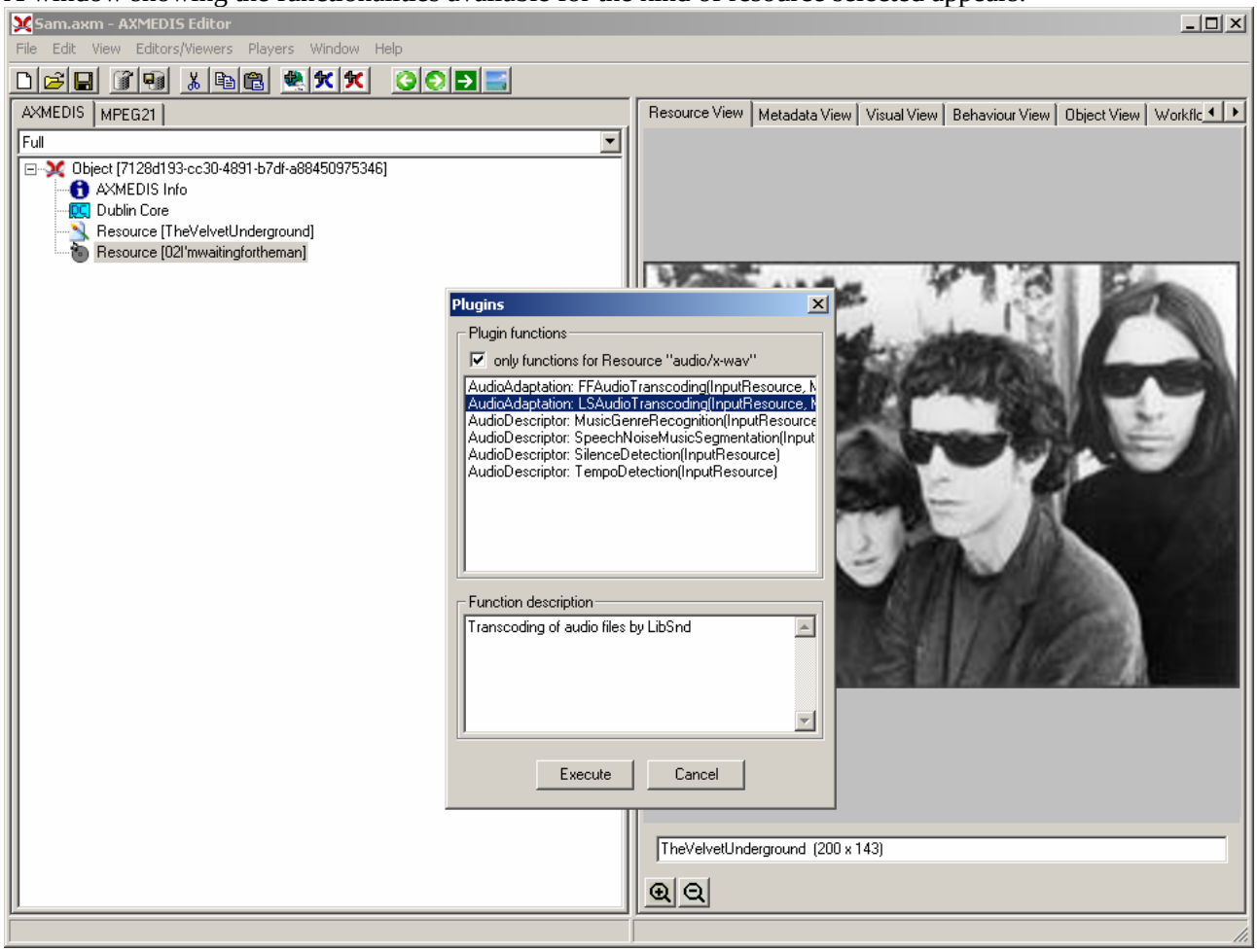
7.3.2 Draft User Manual

Here's an example on how to use the libsndfile audio adaptation transcoding function as a plug-in with for the AXMEDIS editor.

The plug-in must be applied on an audio resource of an AXMEDIS object. The adaptation plug-in is called by right-clicking on the interesting resource and selecting the 'Plugin…' command:



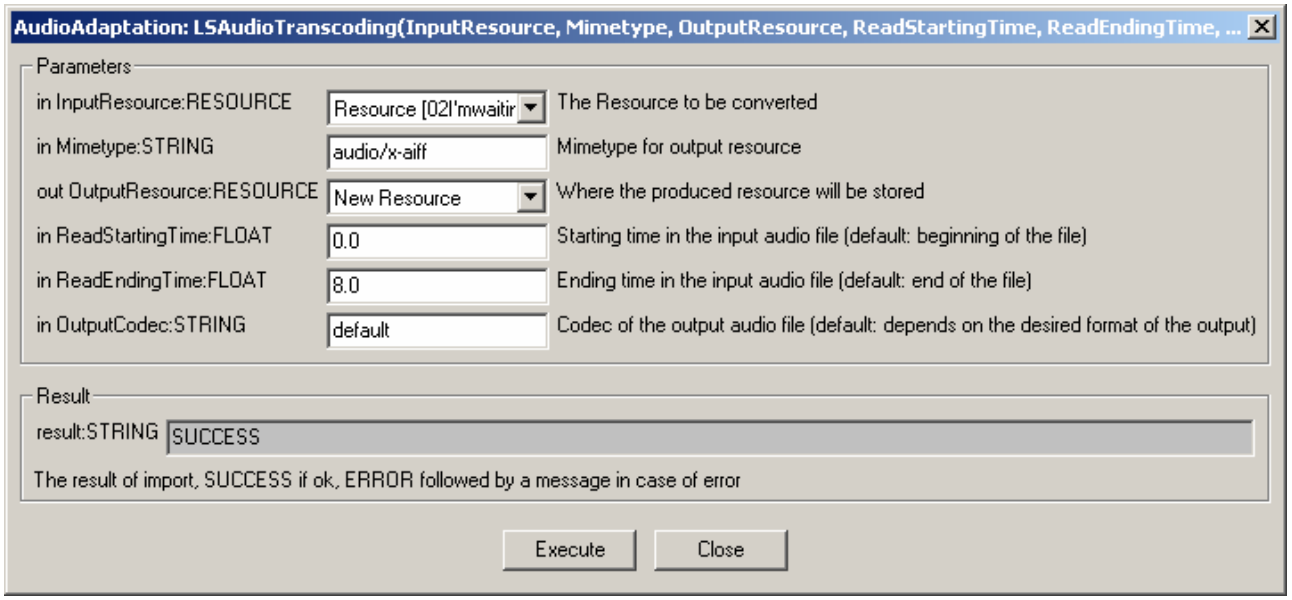
A window showing the functionalities available for the kind of resource selected appears:



The first audio adaptation function available is the libsndfile transcoding function which is selected by clicking on **LSAudioAdaptation: LSAudioTranscoding**. A new window appears showing the interface to the audio transcoding function. In the example of the following figure, the transcoding function is used to create a 10 second snapshot with reduced bit rate of the input audio file:

- AIFF format
- Only a portion of 8 seconds of the input resource is selected (just the beginning of the sound track)

Such a snapshot could be useful for small audio sampling.



7.3.3 Formal description of algorithm libsndfile Transcoding

Description: encode an audio file in another format or another codec and change its sample rate and number of audio channels if needed.

Signature:

string Trancoding(*AxResource* InputResource, *string* MimeType, *AxResource* OutputResource, *float* ReadStartingTime, *float* ReadEndingTime, *string* OutputCodec)

Parameter List:

Name: InputResource
Description: the resource to be converted
Parameter Type: *AxResource*
Default Value:
Constraints:
Resource Type: audio
Resource Format: x-mpeg (.mp3), x.aiff (.aif, .aiff), x-wav (.wav), basic (.au, .snd), x-ms-wma (.wma), x-vorbis (.ogg), x-pn-realaudio (.ra, .ram)
Ranges:

Name: MimeType
Description: MimeType for the output resource
Parameter Type: *string*
Default Value:
Constraints:
Resource Type: audio
Resource Format: x-mpeg, x-aiff, x-wav, basic, x-vorbis, x-ac3
Ranges:

Name: OutputResource
Description: Where the output resource will be stored
Parameter Type: *AxResource*
Default Value:
Constraints:
Range:

Name: ReadStartingTime

Description: set the beginning of the output resource to *ReadStartingTime* seconds from the beginning of the input resource

Parameter Type: *float*

Default Value: by default, the read starting time is set to 0 seconds which means that the input resource is considered from the beginning

Constraints:

Range:

Name: ReadEndingTime

Description: set the end of the output resource at *ReadEndingTime* seconds from the beginning of the input resource

Parameter Type: *float*

Default Value: by default, the read ending time is set to the end of the input resource

Constraints:

Range:

Name: OutputCodec

Description: set the codec of the output resource; depending on the mime type selected for the output resource, only a certain subset of codec will be supported (the following table shows the possible codecs according to the possible mime types)

Parameter Type: *string*

Default Value: the default codec depend on the mime type selected for the output resource (the following table shows the default codec according to the possible mime types)

Constraints:

Range:

Result: Result

Result Type: string

Result Description: the result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

Libsndfile supported types and codecs:

For a list of codecs and formats supported by the Libsndfile library, please refer to section 34.5.

Mime Type accepted :

```
audio/x-wav
audio/x-basic
audio/x-paris
audio/x-svx
audio/x-nist
audio/x-voc
audio/x-ircam
audio/x-w64
audio/x-sd2
audio/x-flac
application/x-pcm
application/x-pagerecall
```

8 Adaptation Tools and Algorithms for Multimedia (EPFL)

Module/Tool Profile Tools and Algorithms for Multimedia Adaptation		
Responsible Name	Mattavelli	
Responsible Partner	EPFL	
Status (proposed/approved)	Proposed	
Implemented/not implemented	Implemented	
Status of the implementation	Plug-In	
Executable or Library/module (Support)	Library	
Single Thread or Multithread	Single Thread	
Language of Development	C++	
Platforms supported	Win32	
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/newrepos/Framework/source/adaptation/multimedia	
Reference to the AXFW location of the demonstrator executable tool for internal download	https://cvs.axmedis.org/newrepos/Framework/bin/adaptation/multimedia	
Reference to the AXFW location of the demonstrator executable tool for public download	--	
Address for accessing to WebServices if any, add accession information (user aNd Passwd) if any	--	
Test cases (present/absent)	absent	
Test cases location	--	
Usage of the AXMEDIS configuration manager (yes/no)	No	
Usage of the AXMEDIS Error Manager (yes/no)	no	
Major Problems not solved	-- --	
Major pending requirements	-- --	
Interfaces API with other tools, named as	Name of the communicating tools References to other major components needed	Communication model and format (protected or not, etc.)
Formats Used	Shared with	format name or reference to a section

Protocol Used	Shared with	Protocol name or reference to a section
Used Database name		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
GPAC	0.4.0	LGPL

8.1.1 General Description of the Module

Text and images have long been the main resources for Web content. Yet, as new formats emerge, rich multimedia presentations are making their entrance into the Web world and are being increasingly used for newscasts, educational material, entertainment etc. At the same time, the 3rd Generation Partnership Project and the 3rd Generation Partnership Project 2 seek to provide uniform delivery of rich multimedia over newly evolved, broadband mobile networks (3rd generation networks) to the latest enabled cell phones.

By rich media, we refer here to a broad range of digital media choreographing audio, video, text, graphics and synthetic animations in real time. Another key feature of such media is interactivity. Rich media may indeed respond directly to user interactions allowing for new ways of consuming media. For example, a pre-recorded webcast may be coupled with a synchronized slide show that allows user interaction with the mouse.

In the recent years, a large number of new multimedia formats have been created independently from each other including among others Quicktime, RealVideo, Advanced Streaming Format, Shockwave, SMIL and MPEG-4 BIFS. AXMEDIS Multimedia Adaptation tools focus on the last two since they are non-uproprietary and support a large variety of usage scenario.

To adapt multimedia files two goals are considered: direct transcoding of a multimedia file towards another multimedia file format and adaptation of the simpler media files embedded into the multimedia file. The later goal asks for means to extract simple media files from multimedia files and means to reintegrate adapted media files into the richer multimedia file. More specifically, the multimedia adaptation tool should provide:

- Transcoding functions: SMIL to MP4, MP4 to ISMA compliant MP4, MP4 to 3GP compliant MP4, SWF to MP4.

- Extraction functions: extraction of media files from richer media files (for example, extracting a video file from a multimedia file to apply specific video adaptation functions).
- Embedding functions: embedding of media files into richer media files (for example, adding an MP3 file into a MP4 file which already contains a video track and a subtitle track).

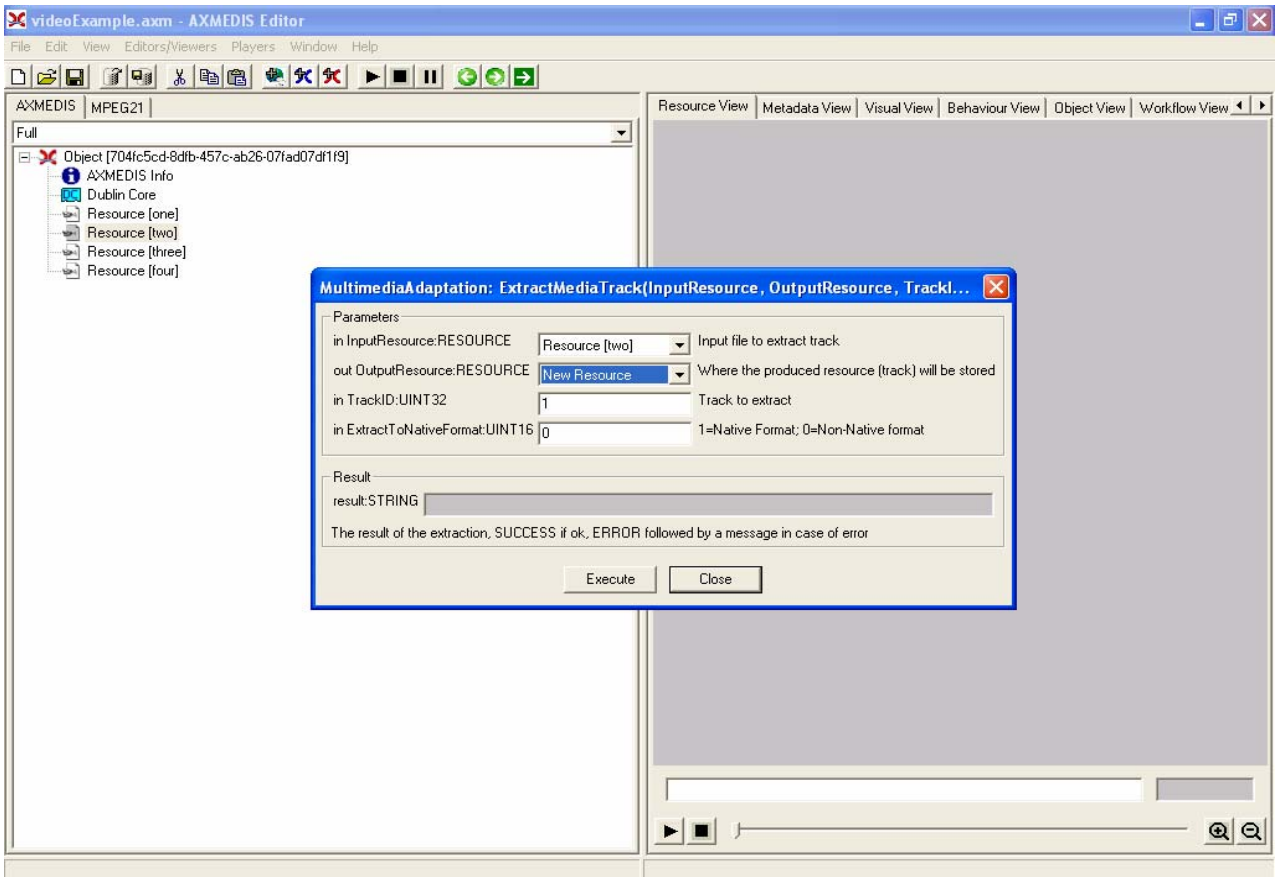
The following subsections describe the external library that may be used in the AXMEDIS framework to implement the needed multimedia adaptation functionalities.

8.1.2 User interface description

The multimedia adaptation functionalities are to be used as plug-ins through the AXCP interface. The interface of AXCP plug-ins maps exactly the formal description of the function and allows entering textually all parameters of the function. Moreover, it displays a brief description of the meaning of the parameters of the function to ease their use. The result of the adaptation is displayed as a textual message in the *Result* box of the interface.

The adaptation is launched by clicking the *Execute* button and the window can be closed with the *Close* button once the adaptation has been performed.

The following figure shows the user interface of the ExtractMediaTrack function. Please refer to section 4.5.10 for the formal description of the ExtractMediaTrack function and to understand how the user interface reflects this formal description.



8.1.3 Technical and Installation information

The multimedia adaptation functionalities are to be used as plug-ins through the AXCP interface. The plug-in simply consists of a DLL and an XML file describing the functionalities of the DLL. Both the DLL and

the XML description should be installed in the plug-in directory of the AXCP compliant tool using the plug-in.

References to other major components needed	NONE
Problems not solved	• NONE
Configuration and execution context	--

8.1.4 Draft User Manual

8.1.4.1 ExtractMediaTrack

This function extracts one track from the original source into a separate file. This extraction can be done in two ways:

- Extraction of the track into an mp4 file with a single track.
- Extraction in the native format of the track (mpeg, mp3…).

8.1.4.2 MP4to3GP

This function translates the MP4 input resource into a new resource with the 3GP format.

8.1.4.3 MP4toISMA

This function translates the MP4 input resource into a new resource conforming to the ISMA specification.

8.1.4.4 CatMultimediaFiles

This function concatenates two whole multimedia resources and produces a new resource containing the concatenation of the initial resources.

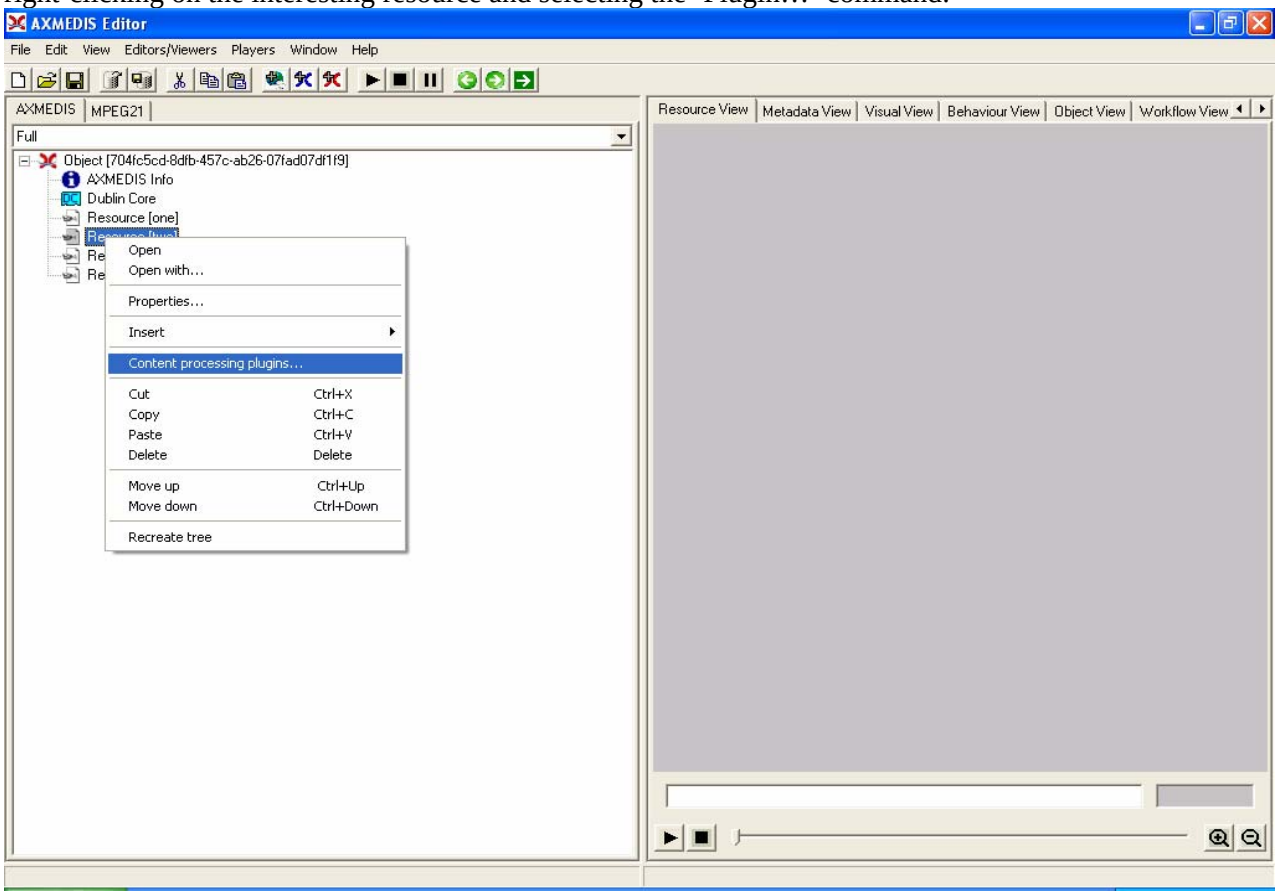
8.1.4.5 AddMediaFiles

This function imports multimedia resources as new tracks into new or already existing mp4 files. It must be specified the size (amount of seconds) of the multimedia resource that is imported and when should it begin inside the destination file, it is to say, the delay of the new track.

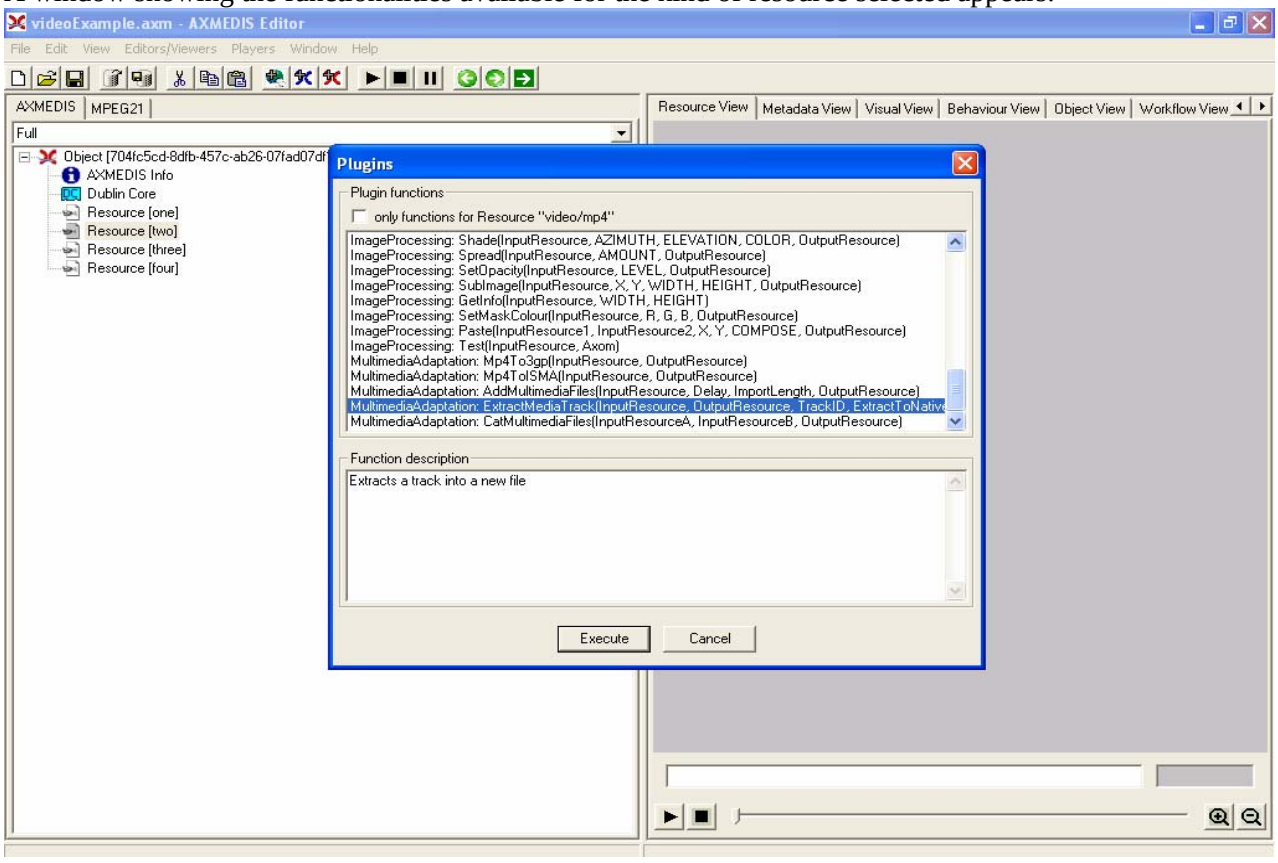
8.1.5 Examples of usage

Here’s an example on how to use the multimedia adaptation track extraction function as a plug-in for the AXMEDIS editor.

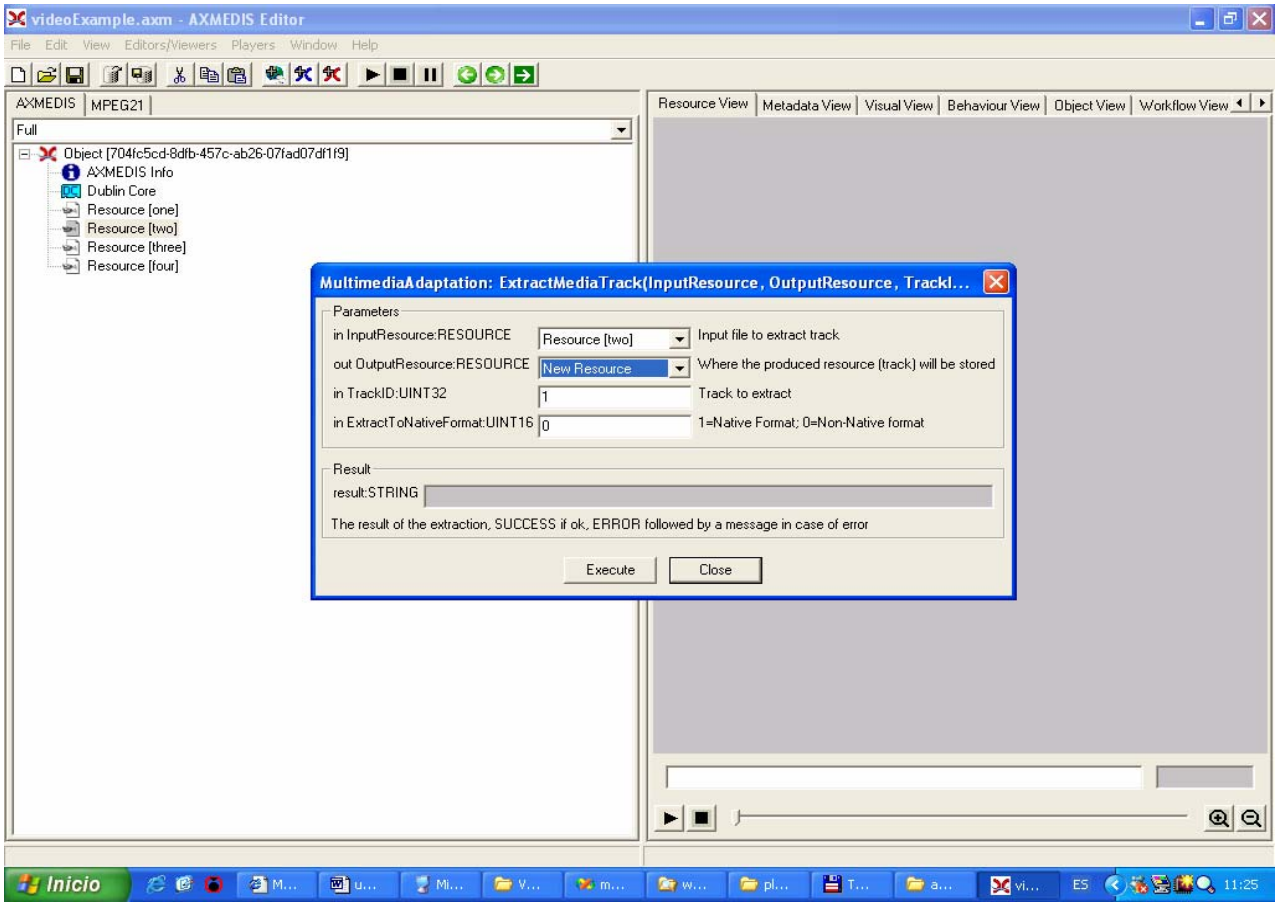
The plug-in must be applied on an MP4 resource of an AXMEDIS object. The adaptation plug-in is called by right-clicking on the interesting resource and selecting the 'Plugin…' command:



A window showing the functionalities available for the kind of resource selected appears:



The ExtractMediaTrack function allows extracting a media track from an input MP4 resource into a new resource. In this example, we extract the first track of the input resource. The first track is an audio track in MP3 format. The extraction is selected to be in non-native format i.e. the extracted media track will not be a MP3 file but it will be converted to MP4. Once extracted, further adaptation can be performed with other plug-ins.



8.1.6 Integration and compilation issues

The tool has been compiled and tested successfully on Win32 platform. It should be ported easily on Linux and MacOsX platforms though it has not been tested yet.

8.1.7 Formal description of algorithm ExtractMediaTrack

Description: extracts one track from the original source into a separate file. This extraction can be done in two ways:

- Extraction of the track into an mp4 file with a single track.
- Extraction in the native format of the track (mpeg, mp3…).

Signature:

string ExtractMediaTrack(*AxResource* InputResource, *AxResource* OutputResource, *UINT32*, TrackID, *UINT16* ExtractToNativeFormat)

Parameter List:

Name: InputResource

Description: the resource to be converted

Parameter Type: *AxResource*

Default Value:

Constraints:
Ranges:

Name: OutputResource
Description: Where the output resource will be stored
Parameter Type: *AxResource*
Default Value:
Constraints:
Range:

Name: TrackID
Description: The number of the track to extract
Parameter Type: *uint32*
Default Value:
Constraints:
Range:

Name: ExtractToNativeFormat
Description: 1=Native format (.mp3, mp2 etc,,,), 0=Non.Native Format (.mp4)
Parameter Type: *uint16*
Default Value:
Constraints:
Range:

8.1.8 Formal description of algorithm MP4o3GP

Description: translates the MP4 input resource into a new resource with the 3GP format.

Signature:
string Mp4To3Gp(*AxResource* InputResource, *AxResource* OutputResource)

Parameter List:
Name: InputResource
Description: the resource to be converted
Parameter Type: *AxResource*
Default Value:
Constraints:
Ranges:

Name: OutputResource
Description: Where the output resource will be stored
Parameter Type: *AxResource*
Default Value:
Constraints:
Range:

8.1.9 Formal description of algorithm MP4toISMA

Description: translates the MP4 input resource into a new resource conforming to the ISMA specification.

Signature:
string Mp4ToIsma(*AxResource* InputResource, *AxResource* OutputResource)

Parameter List:
Name: InputResource
Description: the resource to be converted

Parameter Type: *AxResource*
Default Value:
Constraints:
Ranges:

Name: OutputResource
Description: Where the output resource will be stored
Parameter Type: *AxResource*
Default Value:
Constraints:
Range:

8.1.10 Formal description of algorithm CatMultimediaFiles

Description: concatenates two whole multimedia resources and produces a new resource containing the concatenation of the initial resources.

Signature:
string CatMultimediaFiles(*AxResource* InputResourceA, *AxResource* InputResourceB, *AxResource* OutputResource)

Parameter List:

Name: InputResourceA
Description: the first resource in the concatenation
Parameter Type: *AxResource*
Default Value:
Constraints:
Ranges:

Name: InputResourceB
Description: the second resource in the concatenation
Parameter Type: *AxResource*
Default Value:
Constraints:
Ranges:

Name: OutputResource
Description: Where the output resource will be stored
Parameter Type: *AxResource*
Default Value:
Constraints:
Range:

8.1.11 Formal Descripiton of Algorithm AddMultimediaFiles

Description: This function imports multimedia resources as new tracks into new or already existing mp4 files. It must be specified the size (amount of seconds) of the multimedia resource that is imported and when should it begin inside the destination file, it is to say, the delay of the new track.

Signature:
string AddMultimediaFiles (*AxResource* InputResource, *UINT32* Delay, *UINT32* ImporLength *AxResource* OutputResource)

Parameter List:

Name: InputResource
Description: Resource to be converted
Parameter Type: *AxResource*
Default Value:
Constraints:
Ranges:

Name: Delay
Description: Delay in milliseconds of the new track
Parameter Type: *UINT 32*
Default Value: 0
Constraints:
Ranges:

Name: Delay
Description: Number of seconds to import from input file (starting from the beginning)
Parameter Type: *UINT 32*
Default Value: 0
Constraints:
Ranges:

Name: OutputResource
Description: Where the produced resource will be stored
Parameter Type: *AxResource*
Default Value:
Constraints:
Ranges:

8.2 GPAC

GPAC is a multimedia framework based on the MPEG-4 Systems standard (ISO/IEC 14496-1) developed from scratch in ANSI C. As of version 0.4.0 GPAC is licensed under the GNU Lesser General Public License. Older GPAC versions are available under the GNU General Public License.

The original development goal is to provide a clean (a.k.a. readable by as many people as possible), small and flexible alternative to the MPEG-4 Systems reference software (known as IM1 and distributed in ISO/IEC 14496-5). The MPEG-4 Reference software is indeed a very large piece of software, designed to verify the standard rather than provide a small, production-stable software. GPAC is written in (almost 100% ANSI) C for portability reasons (embedded platforms and DSPs) with a simple goal: keep the memory footprint as low as possible.

The natural evolution has been the integration of recent multimedia standards (SVG/SMIL, VRML, X3D, SWF, 3GPP(2) tools, etc) into a single framework. VRML97 and a good amount of the X3D standard have already been integrated into GPAC, as well as some SVG support and experimental Macromedia Flash support.

The current GPAC release (0.4.0) already covers a very large part of the MPEG-4 standard, and has some good support for 3GPP and VRML/X3D, and features what can probably be seen as the most advanced and robust 2D MPEG-4 Player available worldwide, as well as a decent 3D player - have a look at the screenshots page.

GPAC also features MPEG-4 Systems encoders/multiplexers, publishing tools for content distribution for MP4 and 3GPP(2) files and many tools for scene descriptions (MPEG4<->VRML<->X3D converters, SWF->MPEG-4, etc...).

GPAC is currently running under Windows, Linux platforms and WindowsCE/PocketPC 2002 (2D rendering only, not tested on SmartPhones).

8.3 IBM Toolkit for MPEG-4

The IBM toolkit for MPEG-4 consists of a set of technologies compliant with the MPEG-4 standard. It is implemented as a set of Java classes and APIs which can be used to develop MPEG-4 applications for authoring and playback. It is the main toolset around which the XMT specification has been built, as a case study for compatibility of the different multimedia XML-based languages such as SMIL, XMT and VRML. Since the toolkit is Java-based, the applications built on top of it will run on any platform that supports Java.

Yet, the IBM toolkit for MPEG-4 is released under a commercial licensing scheme and is available only for 90 days in trial license. License costs range from 500$ (1000 toolkits for internal use only) to 5000$ (500 toolkits for internal and/or external use and distribution).

8.4 SMIL to MPEG-4 BIFS conversion

Though SMIL and BIFS have very similar functionalities and target applications, the former appears to be easier to author while the latter seems more suited to broadcasting applications and video-on-demand and offers mechanisms for copy protection and intellectual property management. Moreover, there is a broad potential for BIFS in the PDA sector since these appliances are based on chip sets and not downloadable software.

Consequently, to take advantage of SMIL content in a larger range of applications or simply to benefit from existing SMIL authoring tools to develop BIFS content (such as the SMIL editor developed in the context of AXMEDIS), it seems interesting to convert SMIL presentations to BIFS format.

Yet, no existing tool or library exists to do this conversion, so that a SMIL to MPEG-4 BIFS converter will be developed in the context of AXMEDIS.

9 Adaptation Tools and Algorithms for Metadata (UNIVLEEDS)

Module/Tool Profile Tools and Algorithms for Metadata Adaptation	
Responsible Name	Kia Ng, Minh Thang Dang and Royce Neagle
Responsible Partner	UNIVLEEDS
Status (proposed/approved)	Proposed
Implemented/not implemented	Prototype library implemented
Status of the implementation	Prototype library
Executable or Library/module (Support)	Executable
Single Thread or Multithread	Multi Thread
Language of Development	C++
Platforms supported	Windows
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/newrepos/Framework/source/metadatamapper/ https://cvs.axmedis.org/newrepos/Framework/include/metadatamapper/
Reference to the AXFW location of the demonstrator executable tool for internal download	https://cvs.axmedis.org/newrepos/Framework/bin/metadatamapper/metadatamapper.lib
Reference to the AXFW location of the demonstrator executable tool for public download	Absent
Address for accessing to WebServices if any, add accession information (user aNd Passwd) if any	N/A
Test cases (present/absent)	Absent
Test cases location	
Usage of the AXMEDIS configuration manager (yes/no)	No
Usage of the AXMEDIS Error Manager (yes/no)	No
Major Problems not solved	Not Yet
Major pending	improved mapping algorithm

requirements		
Interfaces API with other tools, named as	Name of the communicating tools References to other major components needed	Communication model and format (protected or not, etc.)
N/A		
Formats Used	Shared with	format name or reference to a section
yes		XSLT, Standard format
Protocol Used	Shared with	Protocol name or reference to a section
N/A		
Used Database name		
N/A		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
yes		wxWidgets, Object Graphics Library, PC,
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
yes	Xerces C++ Parser v-2.7.0	Apache Software Licence, v-2.0
	Xalan-c++ v-1.9	The Apache Software License, v1.1
	wxWidgets-2.4	wxWindows licence (http://www.wxwidgets.org/newlicen.htm)

9.1.1 General Description of the Module

The MetaDataMapper library is used by the metadata mapper editor and the metadata adaption tools to create mapping information between source and target XML metadata documents and to convert a source XML metadata document to a target metadata according to existing mapping information.

When used by the metadata mapper editor, the metadatamapper library builds an XSLT document from the mapping information provided by the GUI

The Mapper uses the Apache Xerces-C parser to parse the input XML documents and build an XSLT document containing mapping information. All libraries used are documented in the Used Libraries section. Xalan is used to transform metadata documents according to an XSLT document.

9.1.2 Xerces: XML parsers in Java and C++ (plus Perl and COM)

Xerces provides XML parsing and generation. The library available for both C++ and Java, implementing the W3C XML and DOM (Level 1 and 2) standards, as well as the de facto SAX (version 2) standard. The parsers are highly modular and configurable. Initial support for XML Schema (draft W3C standard) is also provided.

The libraries feature:

1. Source code, samples, and documentation is provided
2. Programmatic generation and validation of XML
3. Pluggable catalogs, validators and encodings

4. Customizable error handling

Xerces includes three Metadata interfaces making it possible to modify or create metadata automatically and dynamically. The following table shows support to the different programming languages (sources are taken from http://xml.apache.org/, http://xml.apache.org/xerces-c/, http://xml.apache.org/xerces2-j/, http://xml.apache.org/xerces-p/).

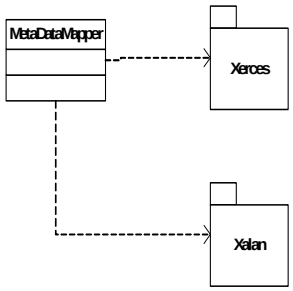
Programming language	Tool/library	Standards Supported
C++	Use Xerces-C++ Version 2.6.0 to read and write XML data. A shared library is provided for parsing, generating, manipulating and validating XML documents	• XML 1.0 (Third Edition) • XML 1.1 (First Edition) (Note: Normalization Checking has not been implemented) • DOM Level 1 Specification • DOM Level 2 Core Specification • DOM Level 2 Traversal and Range Specification • SAX 1.0 and SAX 2.0 • Namespaces in XML • XML Schema Part 1: Structure • XML Schema Part 2: Datatypes • Namespaces in XML 1.1 • DOM Level 3 Core Specification (Partial implementation) • DOM Level 3 Load and Save Specification

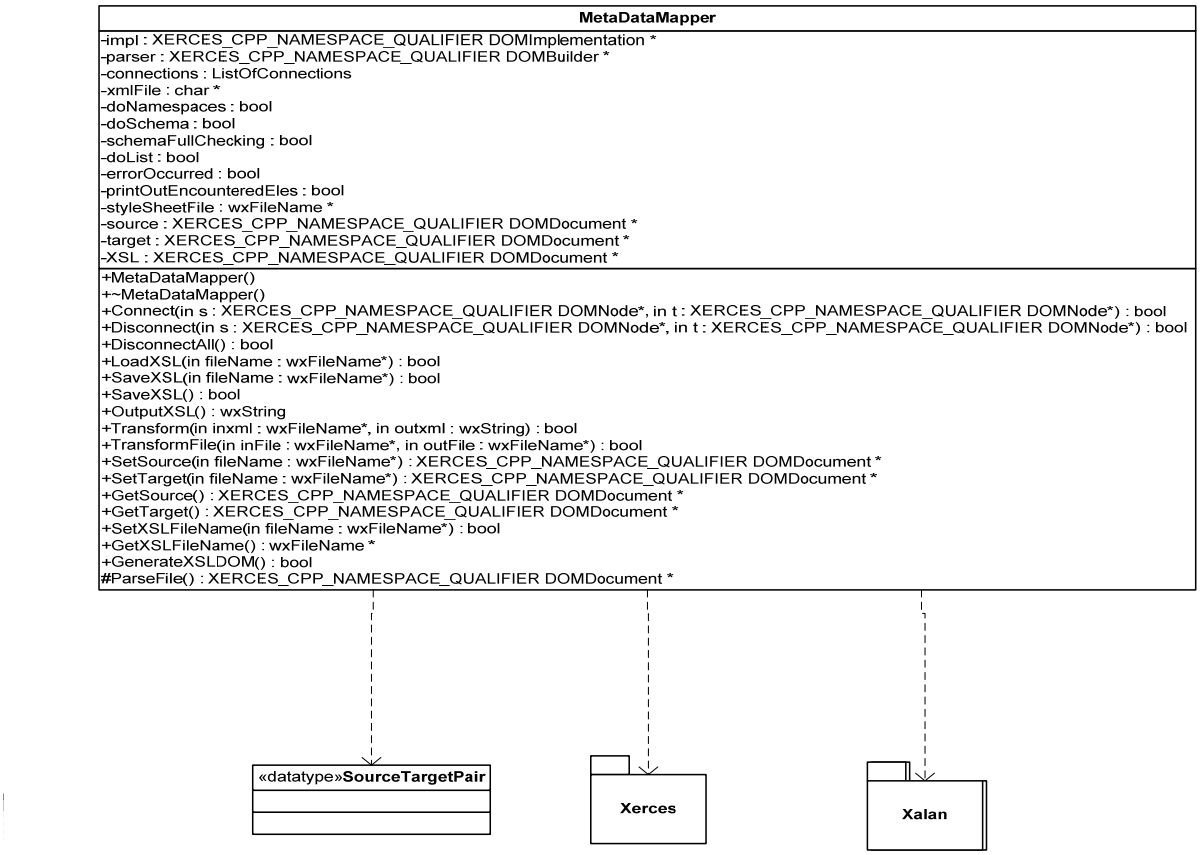
9.1.3 Xalan : XSLT stylesheet processors in Java & C++

Xalan is an XSLT processor for transforming XML documents into HTML, text, or other XML document types. Implementations for XSL Transformations (XSLT) Version 1.0 and the XML Path Language (XPath) Version 1.0, it works with the appropriate Xerces XML parser

Programming language	Tool/library	Description
C++	Use Xalan-C++ Version 1.9 to transform XML documents into HTML, text, or other document types.	• implements XSL Transformations (XSLT) Version 1.0 • XML Path Language (XPath) Version 1.0 • Works with a compatible release of the Xerces-C++ XML parser: Xerces-C++ version 2.6.0

9.1.4 Module Design in terms of Classes





9.1.5 User interface description

The user interface for editing mapping information and using the metadatamapper library to generate XSLT stylesheets is discussed in DE3-1-2-2-4

9.1.6 Technical and Installation information

- is a static library that can be linked in to AXMEDIS tools

Problems not solved	The metadatamapper is currently a prototype. Two main areas need work: 1. The mapping of elements can be improved to handle more complex mapping rules 2. The prototype can only support one layer Metadata tree structure at this stage

9.1.7 Draft User Manual

There is no user manual currently available

9.1.8 Examples of usage

The library is used in the MetadataMapperGUI. First a user loads a source and target metadata file. Next they connect elements to create mapping information. When the user has mapped all the elements that they

require, they save a map file by clicking the save toolbar button. An XSLT file will be saved on the users system which can be used for metadata adaption
Work is also underway to allow adaption of metadata using a JavaScript wrapper around the mapper library. Adaption can be achieved through the GUI using the transform menu item.

9.1.9 Integration and compilation issues

None

9.1.10 Configuration Parameters

Config parameter	Possible values
None	

9.1.11 Errors reported and that may occur

Error code	Description and rationales
5	`out of memory exception when trying to create a new stylesheet`
2	DOM Exception `when trying to create a new stylesheet`
`error in generate xsl function`	`Undefined exception when trying to create a new stylesheet`

9.1.12 Formal description of algorithm to generate XSLT file

This is the first prototype of a simple XSLT generation algorithm. Mappings can be generated between top level elements of the source and target metadata.

```
if (connections.GetCount() >0){
            for(ListOfConnections::Node* node = connections.GetFirst();
                node; node = node->GetNext()){
                SourceTargetPair *temp = node->GetData();

                DOMElement*  foreachElem = XSL-
>createElement(XMLString::transcode("xsl:for-each"));
                foreachElem->setAttribute(
                    XMLString::transcode("select"),
                    temp->sourceNode->getNodeName());
                targetElem->appendChild(foreachElem);
                DOMElement*  tElem = XSL->createElement(temp->targetNode-
>getNodeName());
                          foreachElem->appendChild(tElem);
                DOMElement*  valueElem = XSL-
>createElement(XMLString::transcode("xsl:value-of"));
                valueElem->setAttribute(
                    XMLString::transcode("select"),
                    XMLString::transcode("."));
                tElem->appendChild(valueElem);

            /**/
            }
        }
```

9.2 Adapting AXInfo, Dublin Core, etc. (via XSLT)

A set of mapping files (in XSLT) can be maintained using the Metadata Mapper GUI to support several standards Metadata formats including Dublin Core.

9.3 Loading Metadata Maps

For this section of the implementation, the XSLT is used to specify current mapping information given by the connections in the graphical interface. Alternatively, a stylesheet can be loaded from file and used for transformations.

To generate an XSLT stylesheet from connected nodes in the graphical interface, the MetaDataMapper Connect and Disconnect methods are called within the event handlers of the GUI. Connection and Disconnection events pass a pointer to the source and target DOMNodes as parameters to the methods and the Mapper maintains a list of current connections. The list is a wxList with each node containing a struct of DOMNode pointer pairs, i.e.

```
/// data structure for holding connection pairs between source and
target XML languages
typedef struct {
    XERCES_CPP_NAMESPACE_QUALIFIER DOMNode *sourceNode;
    XERCES_CPP_NAMESPACE_QUALIFIER DOMNode *targetNode;
}SourceTargetPair;

/// array for connection info
WX_DECLARE_LIST(SourceTargetPair,      ListOfConnections);
```

The XSLT DOM Document member is updated each time a change occurs so it always accurately represents the current mapping information. This means the current XSLT could be displayed in a graphical interface to help the user to see the result of connections between nodes.

9.4 Metadata Mapping batch processing

The MetaDataMapper library can be used for adaptation of metadata documents. We are currently working on a number of interfaces to the underlying library. These may include:

1. a graphical user interface for batch processing of metadata documents
2. a command line interface to the same tool
3. a JavaScript wrapper to enable runtime scripting of adaptation library methods

The graphical interface will allow users to select a group of files and process them according to a selected style sheet. The output files will then be written depending on information provided by the user. The user can specify a directory to write the files to (keeping their original name) or a prefix/postfix may be added to the output files in order make it clear they are the result of processing and to avoid overwriting the input files.
The batch processing tool will then take each file and transform it according to the XSLT using the XALAN library.
When run from the command line, the batch processing tool will take a number of command line options.

Command prompt options

-pre the prefix to apply to output filenames
-pos the postfix to apply to output filenames (before the file extension)
-dir output directory (defaults to current working directory)

10 Adaptation Tools and Algorithms for DRM information (UPC)

Module/Tool Profile Tools and Algorithms for DRM Adaptation		
Responsible Name	Rubén Barrio / Xavier Maroñas	
Responsible Partner	UPC	
Status (proposed/approved)	Proposed	
Implemented/not implemented	Implemented	
Status of the implementation	Partially implemented	
Executable or Library/module (Support)	Executable	
Single Thread or Multithread	Multithread	
Language of Development	C++	
Platforms supported	MS Windows	
Reference to the AXFW location of the source code demonstrator	N/A	
Reference to the AXFW location of the demonstrator executable tool for internal download	N/A	
Reference to the AXFW location of the demonstrator executable tool for public download	N/A	
Address for accessing to WebServices if any, add accession information (user and Passwd) if any	N/A	
Test cases (present/absent)	Absent	
Test cases location		
Usage of the AXMEDIS configuration manager (yes/no)	No	
Usage of the AXMEDIS Error Manager (yes/no)	No	
Major Problems not solved		
Major pending requirements		
Interfaces API with other tools, named as	Name of the communicating tools References to other major components needed	Communication model and format (protected or not, etc.)
Formats Used	Shared with	format name or reference to a section
Protocol Used	Shared with	Protocol name or reference to a section

Used Database name		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
Xerces	Xerces C++ Parser v-2.6.0	Apache Software Licence, v-2.0
wxWidgets	wxWindows-2.4.2	

10.1.1 General Description of the Module

DRM adaptation involves the adaptation of the related licenses, as derived AXMEDIS objects or digital resources within the AXMEDIS objects can be seen as new creations with regard to original ones. Therefore, new licenses must be created during the adaptation process, always respecting the terms and conditions fixed in the original license or licenses for the adapted AXMEDIS objects or contents within these objects.

Nevertheless, DRM information (mainly licenses and PARs) inside the AXMEDIS project, that are related to AXMEDIS Objects will be expressed in XML language by using MPEG-21 REL.

In order to adapt this information to different rights expression languages, also based in XML or to adapt a license to be more compact in order to use it into portable devices (for instance, mobile phones or PDAs), we will make use of existing libraries for manipulating XML documents.

The main adaptation function produced by this module can be summarised in:

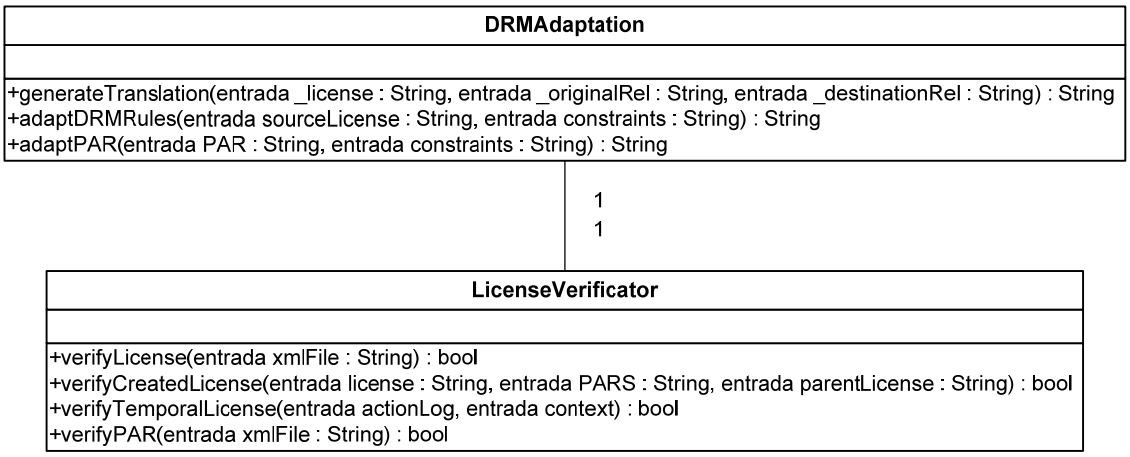
- Compacting licenses for their use in portable devices
- Translating licenses from one rights expression language to another
- Automatic generation of a license when an adaptation over the content it applies is done

For XML DRM rules transcoding, the *Xerces Libraries* can be used to parse a given piece of XML data.

At the present moment, the adaptations foreseen regarding the current state of the art are between MPEG-21 REL and OMA DRM REL (which is based on ODRL) and MPEG-21 REL profiles. Evolution in the state of the art may involve more adaptations available.

10.1.2 Architecture of the module

The following figure shows the UML diagram of the Tools and Algorithms for DRM adaptation module.



Class diagram for tools and algorithms for DRM Adaptation

10.1.3 Module Design in terms of Classes

In order to optimize the components implemented inside AXMEDIS, DRMAdaptation will make use of the license verification module (described in detail in DE3.1.2.2.14) in order to check if the generated adaptation is correct.

10.1.4 Formal description of DRMAdaptation

generateTranslation	
Method	GenerateTranslation
Description	This method translates a license expressed in XML language to another rights expression language according to the parameters passed. The result is validated using license verificator module
Input parameters	String _license String _originalREL String _destinationREL
Output parameters	String resultLicense

adaptDRMRules	
Method	AdaptDRMRules
Description	This method adapts some DRM Rules according to the constraints passed as parameters. The result is a new series of DRM Rules. The result is validated using license verificator module
Input parameters	String sourceLicense String constraints
Output parameters	String resultDRMRules
AdaptPAR	
Method	AdaptPAR
Description	This method adapts some PAR according to the constraints passed as parameters. The result is a new series of PAR. The result is validated using license verificator module
Input parameters	String PAR String constraints
Output parameters	String resultPAR

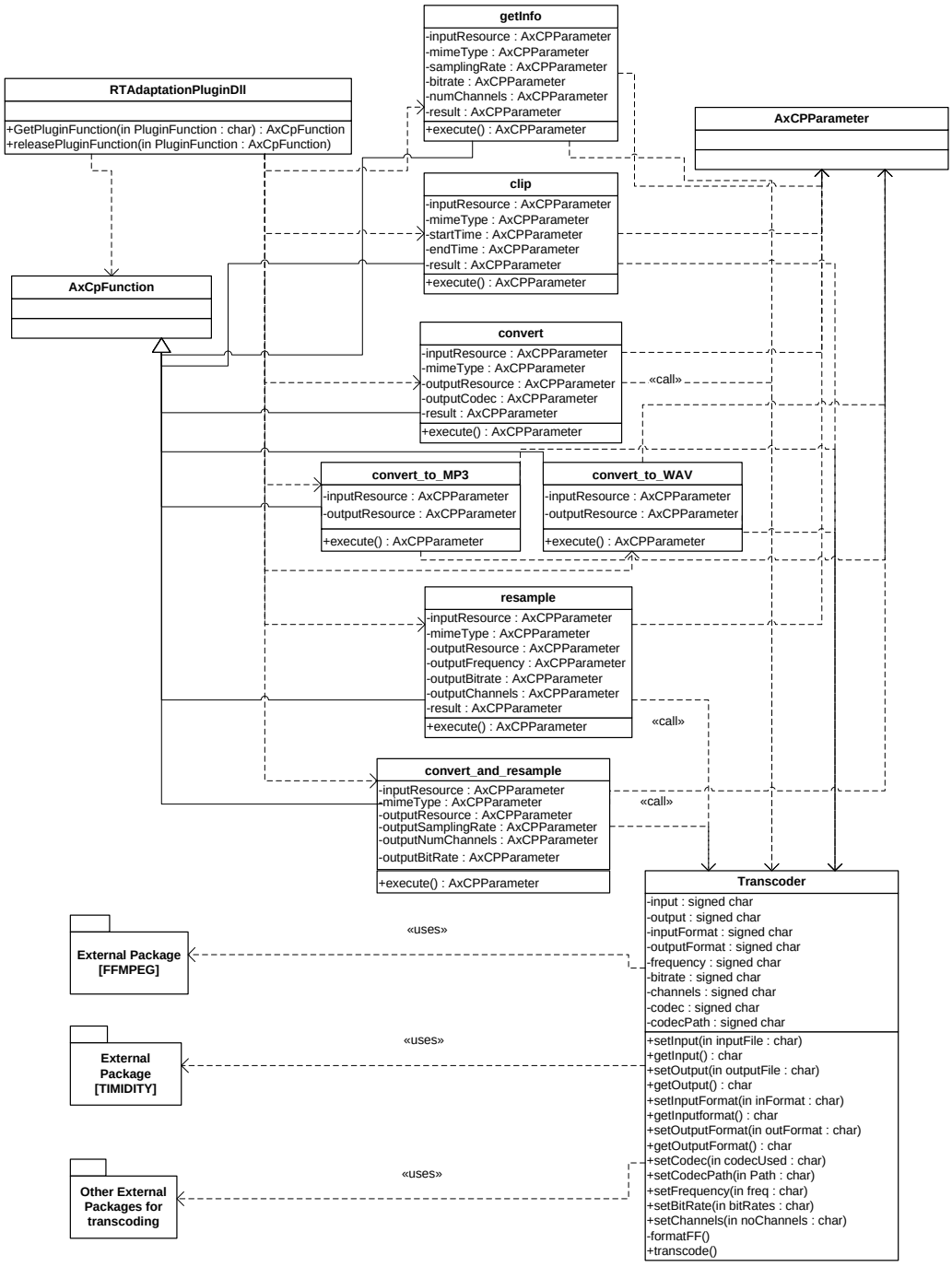
11 Adaptation Tools and Algorithms for Ringtones (UR)

Module/Tool Profile Tools and Algorithms for Ringtone Adaptation		
Responsible Name	Badii	
Responsible Partner	UR	
Status (proposed/approved)	Approved	
Implemented/not implemented	Implemented	
Status of the implementation	First Prototype	
Executable or Library/module (Support)	Library	
Single Thread or Multithread	Multithread	
Language of Development	C++	
Platforms supported	Win32	
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/repos/Framework/src/adaptation/ringtone	
Reference to the AXFW location of the demonstrator executable tool for internal download	https://cvs.axmedis.org/repos/Framework/bin/adaptation/ringtone	
Reference to the AXFW location of the demonstrator executable tool for public download		
Address for accessing to WebServices if any, add accession information (user aNd Passwd) if any		
Test cases (present/absent)	Absent	
Test cases location	http://////////////////	
Usage of the AXMEDIS configuration manager (yes/no)	No	
Usage of the AXMEDIS Error Manager (yes/no)	No	
Major Problems not solved	-- --	
Major pending requirements	-- --	
Interfaces API with other tools, named as	Name of the communicating tools References to other major components needed	Communication model and format (protected or not, etc.)
Formats Used	Shared with	format name or reference to a section

Protocol Used	Shared with	Protocol name or reference to a section
Used Database name		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
FFMPEG		LGPL
LIBSNDFILE		LGPL

11.1.1 General Description of the Module

Ringtone Adaptation refers to the adaptation of ringtones of popular formats to enhance usability and manage the variable delivery to cater for heterogeneous client devices and user requirements on-demand. It uses external libraries like FFMPEG and LIBSNDFILE to convert the ring tones from one format to another and to resample it based on the client device.



11.1.2 User interface description

The ringtone adaptation functionalities are to be used as plug-ins through the AXCP interface. The interface of AXCP plug-ins maps exactly the formal description of the functions of ringtone adaptation, allowing text-based provision of function parameters. Moreover, it displays a brief description of the meaning of the parameters of the function to ease their use. The result of the adaptation is displayed as a text message in the *Result* box of the interface.

The adaptation is launched by clicking the *Execute* button and the window can be closed with the *Close* button once the adaptation has been performed.

11.1.3 Technical and Installation information

References to other major components needed	AxCPParameter, AxCPFunction
Problems not solved	
Configuration and execution context	

11.1.4 Draft User Manual

11.1.4.1 Convert Function

Description: Convert a ringtone to different formats.

Signature:
STRING convert (RESOURCE InputResource, STRING Mimetype, RESOURCE OutputResource)

Parameter List:

Name: InputResource
Description: The Resource to be converted
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: audio
Resource Format: x-wav x-aiff x-ms-wma basic mpeg mid
Ranges:

Name: Mimetype
Description: Mimetype for output resource
Paramater Type *string*
Default Value:
Constraints:

Name: OutputResource
Description: Where the produced resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:

Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

11.1.4.2 Convert_to-MP3 Function

Description: Used to convert a ringtone to MP3 format. The input formats supported currently are x-mpeg (.mp3), x.aiff (.aif, .aiff), x-wav (.wav), basic (.au, .snd), x-ms-wma (.wma), x-vorbis (.ogg), x-pn-realaudio (.ra, .ram)

Signature:
STRING convert_to_MP3 (RESOURCE InputResource, RESOURCE OutputResource)

Parameter List:

Name: InputResource
Description: The Resource to be converted
Paramater Type *AxResource*
Default Value:
Constraints:

Resource Type: audio
Resource Format: x-mpeg (.mp3), x.aiff (.aif, .aiff), x-wav (.wav), basic (.au, .snd), x-ms-wma (.wma), x-vorbis (.ogg), x-pn-realaudio (.ra, .ram)
Ranges:

Name: **OutputResource**
Description: Where the produced resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

11.1.4.3 Convert_to_WAV Function

Description: Used to convert a ringtone to WAV format. The input formats supported currently are x-mpeg (.mp3), x.aiff (.aif, .aiff), x-wav (.wav), basic (.au, .snd), x-ms-wma (.wma), x-vorbis (.ogg), x-pn-realaudio (.ra, .ram)

Signature:
STRING convert_to_WAV (RESOURCE InputResource, RESOURCE OutputResource)

Parameter List
Name: InputResource
Description: The Resource to be converted
Paramater Type *AxResource*
Default Value:
Constraints:
Resource Type: audio
Resource Format: x-mpeg (.mp3), x.aiff (.aif, .aiff), x-wav (.wav), basic (.au, .snd), x-ms-wma (.wma), x-vorbis (.ogg), x-pn-realaudio (.ra, .ram)
Ranges:

Name: OutputResource
Description: Where the produced resource will be stored
Paramater Type *AxResource*
Default Value:
Constraints:
Result: Result
Result type: *string*
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

11.1.4.4 Resample Function

Description: Resample the input file (i.e. changing frequency, bitrate etc)

Signature:
STRING resample (RESOURCE InputResource, STRING Mimetype, RESOURCE OutputResource, UINT32 OutputSamplingRate, UINT16 OutputNumChannels, UINT16 OutputBitRate)

Parameter List

Name: InputResource
Description: The Resource to be converted
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: audio
Resource Format: x-wav x-aiff x-ms-wma basic mpeg mid
Ranges:

Name: Mimetype
Description: Mimetype for output resource
Paramater Type STRING
Default Value:
Constraints:

Name: OutputResource
Description: Where the produced resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:

Name: OutputSamplingRate
Description: Sampling rate of the output audio file (default: sampling rate of the input)
Paramater Type UINT32
Default Value:
Constraints:
Resource Type:
Ranges:

Name: OutputNumChannels
Description: Number of channels of the output audio file (default: number of channels of the input)
Paramater Type UINT16
Default Value:
Constraints:
Resource Type:
Ranges:

Name: OutputBitRate
Description: Bit rate of the output audio file - Only applies to compressed audio file formats (default: 64 kb)
Paramater Type UINT16
Default Value:
Constraints:
Resource Type:
Ranges:

Result: Result
Result type: STRING
Result Description: The result of import, SUCCESS if ok, ERROR followed by a message in case of error

11.1.4.5 Convert_And_Resample Function

Description: Converts the file into any supporting formats and resample it (i.e. changing frequency, bitrate, sampling rate etc) at the same time. (Please note that some values of the Sampling rates and frequencies can't exist together according to the ffmpeg library used and hence if the plugin shows unknown exception then please restart the plugin and give different values)

Signature:

STRING convert_and_resample (RESOURCE InputResource, STRING MimeType, RESOURCE OutputResource, UINT32 OutputSamplingRate, UINT16 OutputNumChannels, UINT16 OutputBitRate)

Parameter List:

Name: InputResource
Description: The Resource to be converted
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: audio
Resource Format: x-mpeg (.mp3), x.aiff (.aif, .aiff), x-wav (.wav), basic (.au, .snd), x-ms-wma (.wma), x-vorbis (.ogg), x-pn-realaudio (.ra, .ram)
Ranges:

Name: Mimetype
Description: Mimetype for output resource
Paramater Type STRING
Default Value:
Constraints:
Resource Type: audio
Resource Format: x-mpeg, x-aiff, x-wav, basic, x-vorbis, x-ac3

Name: OutputResource
Description: Where the produced resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:

Name: OutputSamplingRate
Description: Sampling rate of the output audio file (default: sampling rate of the input)
Paramater Type UINT32
Default Value:
Constraints:
Resource Type:
Ranges:

Name: OutputNumChannels
Description: Number of channels of the output audio file (default: number of channels of the input)
Paramater Type UINT16
Default Value:
Constraints:
Resource Type:
Ranges:

Name: OutputBitRate
Description: Bit rate of the output audio file - Only applies to compressed audio file formats (default: 64 kb)
Paramater Type UINT16
Default Value:
Constraints:
Resource Type:
Ranges:

Result: Result
Result type: STRING

Result Description: The result of import, SUCCESS if ok, ERROR followed by a message in case of error

11.1.4.6 Function getInfo

Description: Get all the information about the input Ring Tone
Signature:
STRING getInfo (RESOURCE InputResource, STRING Mimetype, UINT32 SamplingRate, UINT16 NumChannels, UINT16 BitRate)
Parameter List

Name: InputResource
Description: The Resource to be converted
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: audio
Resource Format: x-wav x-aiff x-ms-wma basic mpeg mid
Ranges:

Name: Mimetype
Description: Mimetype for output resource
Paramater Type STRING
Default Value:
Constraints:

Name: SamplingRate
Description: Sampling rate of the input ring tone
Paramater Type UINT32
Default Value:
Constraints:
Resource Type:
Ranges:

Name: NumChannels
Description: Number of channels of the input ring tone
Paramater Type UINT16
Default Value:
Constraints:
Resource Type:
Ranges:

Name: BitRate
Description: Bit rate of the input ring tone - (default: 64 kb)
Paramater Type UINT16
Default Value:
Constraints:
Resource Type:
Ranges:

Result: Result
Result type: STRING
Result Description: The result of import, SUCCESS if ok, ERROR followed by a message in case of error

11.1.4.7 Function clip

Description: Clip the file for the specified time (for e.g. reducing it to a 30 sec clip)

Signature:
STRING clip (RESOURCE InputResource, STRING Mimetype, FLOAT ReadStartingTime, FLOAT ReadEndingTime)

Parameter List

Name: InputResource
Description: The Resource to be converted
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: audio
Resource Format: x-wav x-aiff x-ms-wma basic mpeg mid
Ranges:

Name: Mimetype
Description: Mimetype for output resource
Paramater Type STRING
Default Value:
Constraints:

Name: ReadStartingTime
Description: Starting time for the clip(default: beginning of the file)
Paramater Type FLOAT
Default Value:
Constraints:
Resource Type:
Ranges:

Name: ReadEndingTime
Description: Ending time for the clip (default: end of the file)
Paramater Type FLOAT
Default Value:
Constraints:
Resource Type:
Ranges:

Result: Result
Result type: STRING
Result Description: The result of import, SUCCESS if ok, ERROR followed by a message in case of error

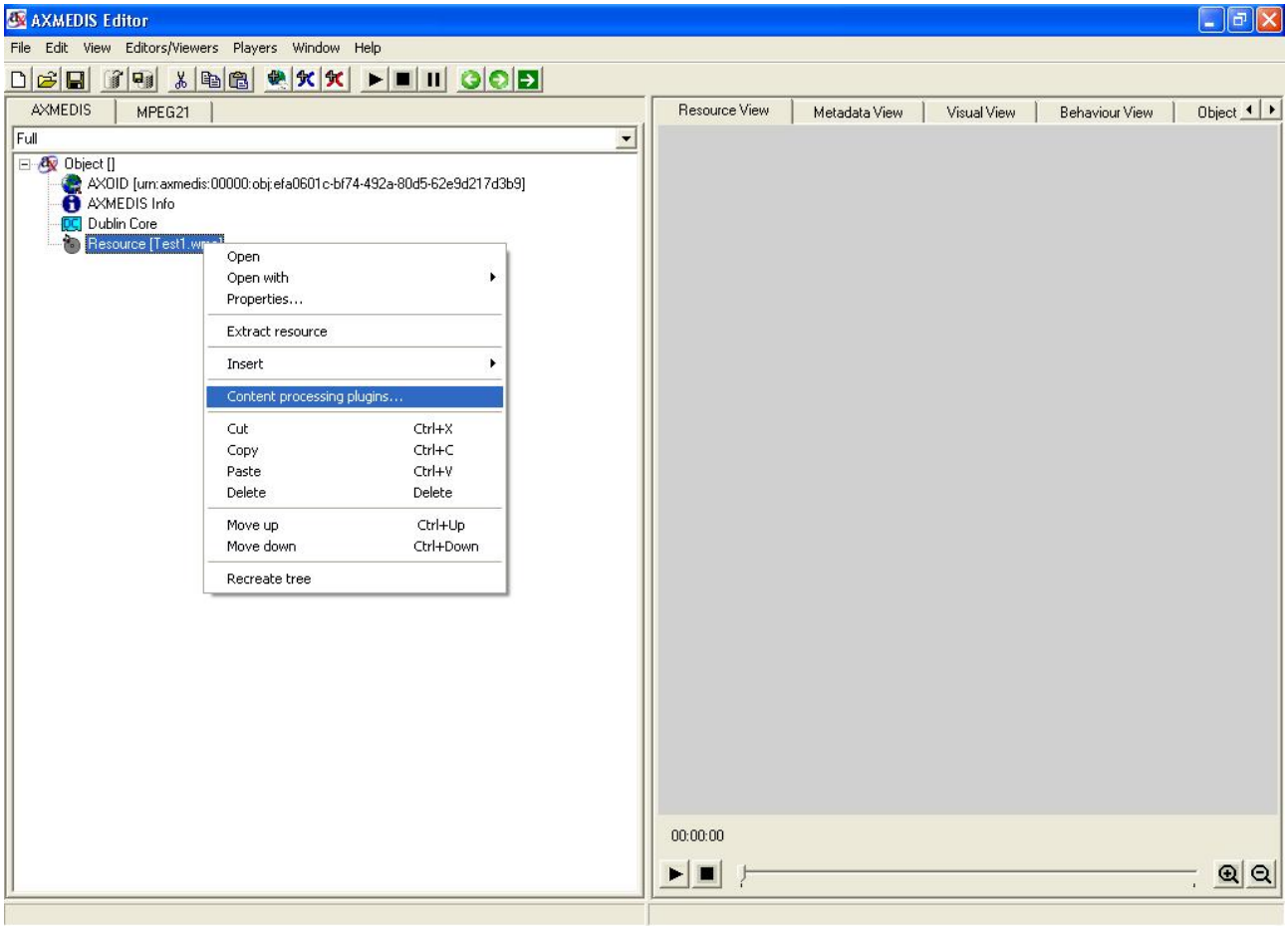
The functionality implemented is based on the following libraries (details can be found in the appendix):

- FFMPEG
- LIBSNDFILE

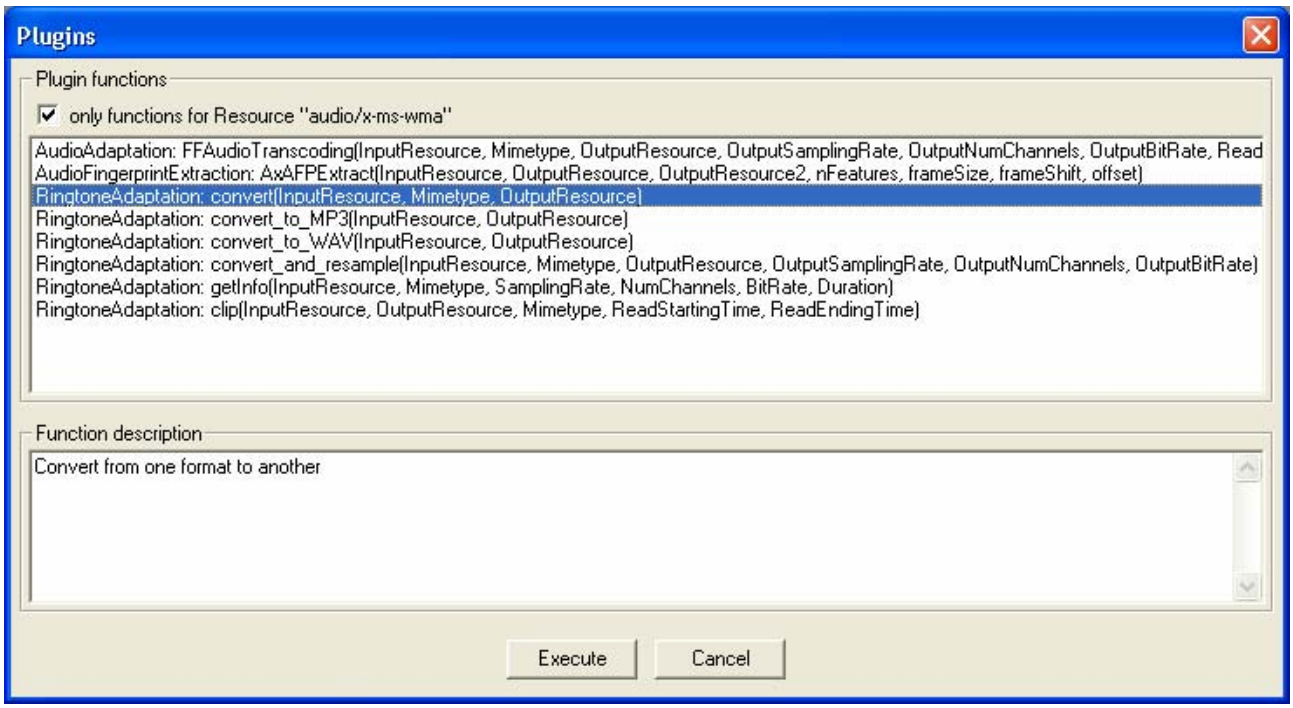
11.1.5 Examples of usage

Here's an example on how to use the plug-in with the AXMEDIS Editor.

Load an embedded resource (audio/ringtone file) into the AXMEDIS Editor. Right click on the resource and select Content processing plugins.

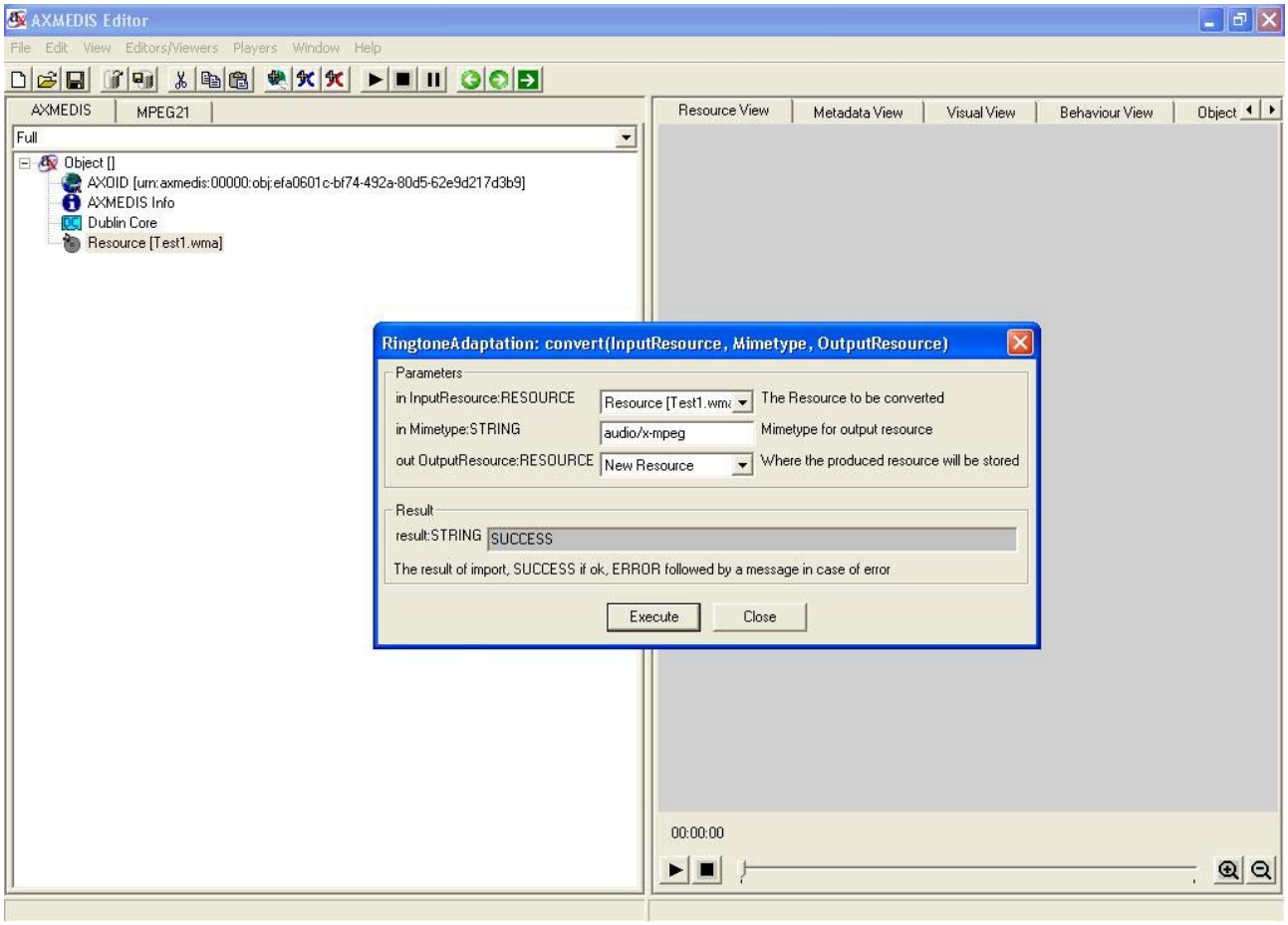


A window will pop up showing the different content processing plugins available for the particular resource, in our case it is ringtone.



Select the Convert function to convert the ringtone to any popular format. The formats supported are x-mpeg (.mp3), x.aiff (.aif, .aiff), x-wav (.wav), basic (.au, .snd), x-ms-wma (.wma), x-vorbis (.ogg), x-pn-realaudio (.ra, .ram)

It will take you to the next screen where you can specify the various parameters for converting the ringtone. Once you enter the parameters and click execute, it will convert the ringtone to the appropriate format. If the ringtone conversion is successful then in result's space you can see SUCCESS or else it will return Error along with an error message.



12 Descriptor extractor as fingerprint for Text files (DIPITA)

Module/Tool Profile Descriptor extractor as fingerprint for Text files		
Responsible Name	Fabbri	
Responsible Partner	DIPITA	
Status (proposed/approved)	Proposed	
Implemented/not implemented	Implemented	
Status of the implementation	70%	
Executable or Library/module (Support)	Library	
Single Thread or Multithread	Single	
Language of Development	C++	
Platforms supported	Win32	
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/newrepos/Framework/{source,include,project}/descriptor/document/	
Reference to the AXFW location of the demonstrator executable tool for internal download	https://cvs.axmedis.org/newrepos/Framework/bin/descriptor/document/	
Reference to the AXFW location of the demonstrator executable tool for public download		
Address for accessing to WebServices if any, add accession information (user aNd Passwd) if any		
Test cases (present/absent)	absent	
Test cases location		
Usage of the AXMEDIS configuration manager (yes/no)	no	
Usage of the AXMEDIS Error Manager (yes/no)	no	
Major Problems not solved	-- --	
Major pending requirements	Linguistic resources for languages other than English Licenses issues of non-free linguistic tools and resources (TreeTagger, non-English Wordnet,…)	
Interfaces API with other tools, named as	Name of the communicating tools References to other major components needed	Communication model and format (protected or not, etc.)
Formats Used	Shared with	format name or reference to a section

Protocol Used	Shared with	Protocol name or reference to a section
Used Database name		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
WordNet (English, Italian, Spanish, French, German)		Free for English, proprietary for other languages
TreeTagger		Free for research. Proprietary for commercial use. Not redistributable.
WordNet Domains		Free for research. Proprietary for commercial use.
Boost		Open source and free for any purpose.

12.1.1 General Description of the Module

Automatic keyword extraction aims to provide the user with a set of descriptors which represent the contents of the document, and which are further exploited to perform advanced searches within the textual repository. Extracted keywords can also be used to identify higher levels of descriptors, such as the document domain.
The procedure is divided into three main steps:

1) Comparative frequency analysis: mono-term keywords extraction.

This process is performed through an integration of resources and standard algorithms, by means of comparison between the document and the referring universe, represented by a general corpus. The output is a list of nouns in the document, ordered by TF.IDF scores

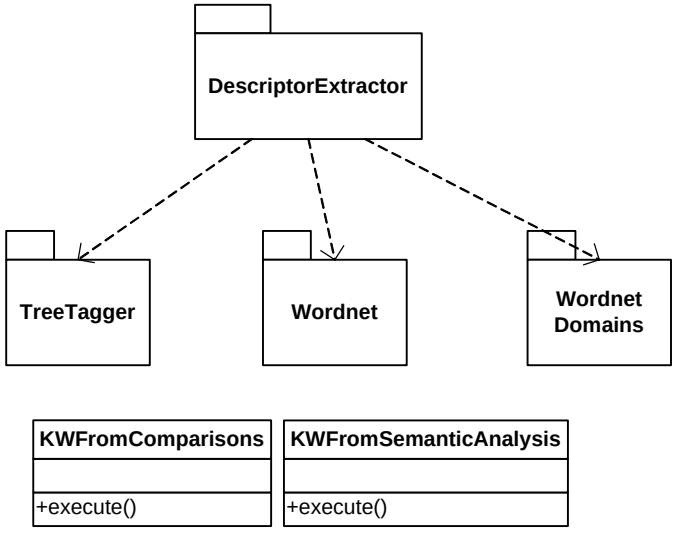
2) Semantic analysis: mono-term keywords semantic disambiguation and domain detection.

Since nouns are potentially ambiguous with respect with their semantics, a word sense disambiguation (WSD) procedure is run over the output of the previous step. This would allow, in principle, the translatability of keywords in different languages. WordNet Domains database is also exploited to determine the "area of discussion" to which each keyword belongs, so providing other keys for content identification.

3) Internal analysis of lexical associations: multi-term keywords detection.

The extracted keywords are further refined from the point of view of the accuracy, of the content identification, and of the value of the descriptors, referring to language properties of word association: high frequency collocations within the text are considered more definite and highly representative of its content.

12.1.2 Module Design in terms of Classes



12.1.3 Technical and Installation information

Due to the need of several external tools and resources that were not designed to work together, the installation is a bit tricky. DIPITA is planning to reduce as far as possible the installation problems.

1. Install TreeTagger:

TreeTagger cannot be distributed by AXMEDIS, but it can be freely downloaded for evaluation purposes from Stuttgart University site (http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/).

Download and unzip tree-tagger-windows-3.1.zip (ftp://ftp.ims.uni-stuttgart.de/pub/corpora/tree-tagger-windows-3.1.zip) in C:\ disk and follow TreeTagger install instructions contained in README.install file. English parameter files can be found at ftp://ftp.ims.uni-stuttgart.de/pub/corpora/english-par-linux-3.1.bin.gz

2. Place the following files in the dir of the tool which exploits the plug-in (they are all placed in the descriptor.zip package). For example, if you use axeditor, place them in the same dir as the axeditor executable file one.

- data-noun.txt (must be unzipped)
- index-noun.txt (must be unzipped)
- lesk-relation.dat
- corpora.txt
- bnc-freq.txt (must be unzipped)
- ngramrules.txt
- stopngrams.txt
- stopwords.txt
- wn-domains-3.0beta-20050224-nouns-only.txt
- tag-english.bat

3. Put the following files in the plug-in directory of the tool you are using:

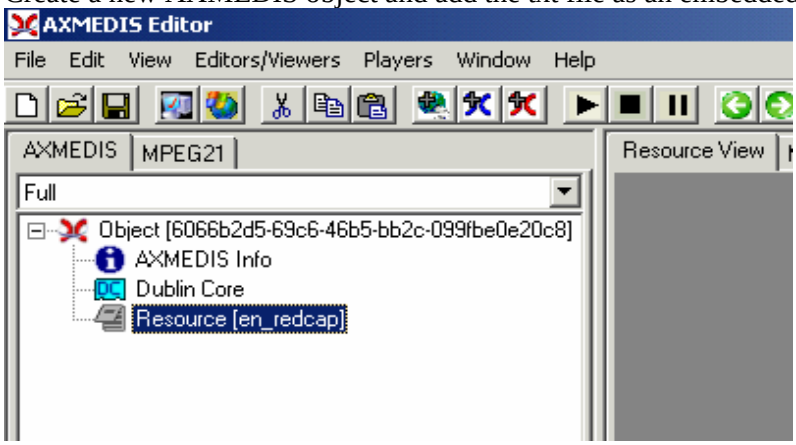
- descriptorextractorplugin.xml
- descriptorextractorplugin.dll

12.1.4 Draft User Manual and examples of usage

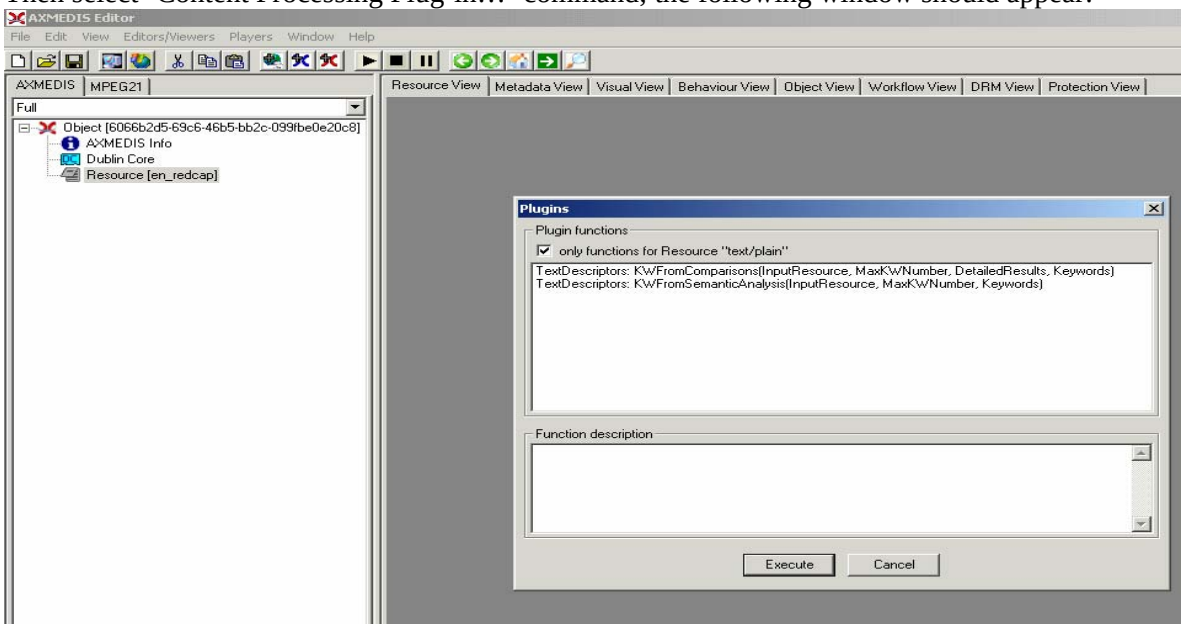
Here's an example on how to use the plug-in with AXEditor.

The plug-in can be applied only to plain text resources and, so far, will give meaningful results only to English texts. In the demonstrator package there is a sample file to test: en_redcap.txt. It's the well known Red Cap tale.

Create a new AXMEDIS object and add the txt file as an embedded resource.



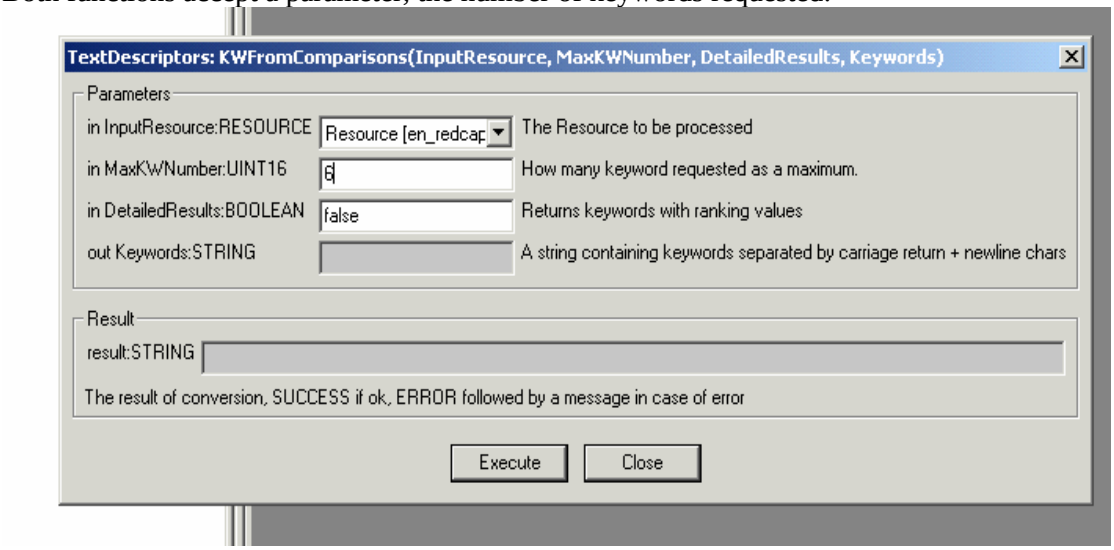
Then select 'Content Processing Plug-in…' command; the following window should appear:



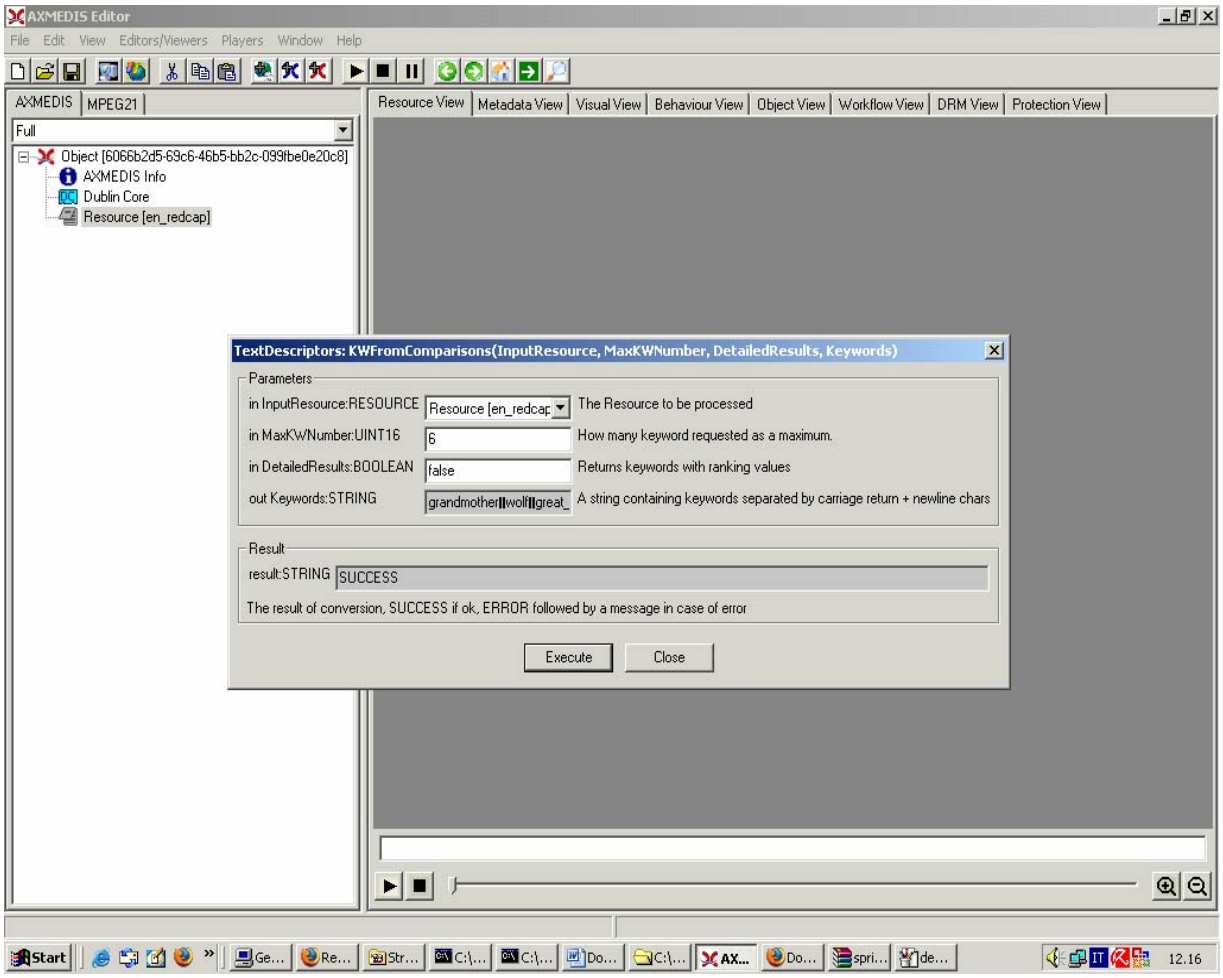
There are 2 functions available:

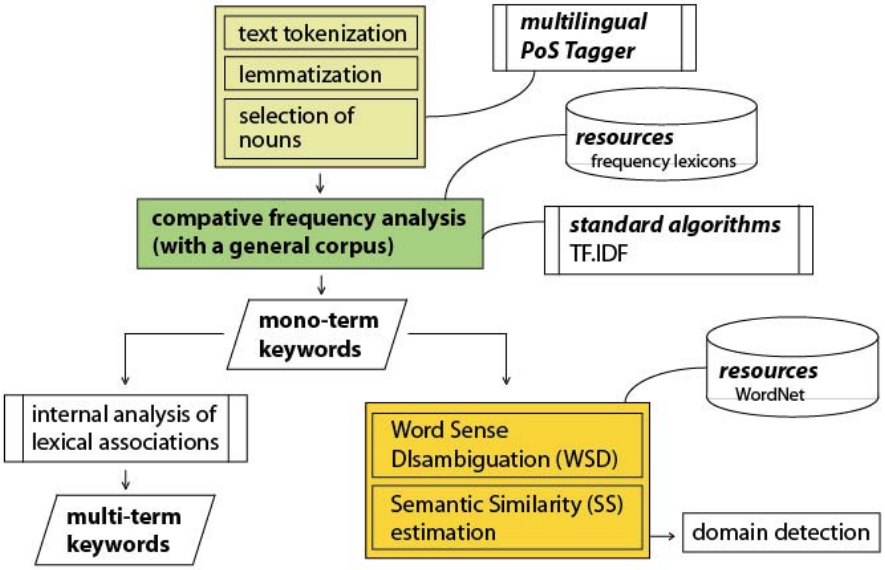
- KWFromComparisons: extracts single and multi-words making a statistical comparison against a reference corpus (British National Corpus);
- KWFromSemanticAnalysis: extracts single and multi-words making a further anlysis with the help of a semantic resource (WordNet).

Both functions accept a parameter, the number of keywords requested:



Clicking on execute makes the plug-in run. Output is given in the 'out' field as a carriage-return separated list of words/multi-words:





12.1.5 Formal description of algorithm

name	
Method	KWFromComparisons
Description	Retrieves the keywords exploiting frequency lists corpus only

Input parameters	RESOURCE:InputResource – The text document from which keywords have to be extracted UINT16:MaxKWNumber – The maximum number of keyword requested
Output parameters	STRING:Keywords – A string containing keywords separated by return + newline chars

name	
Method	KWFromSemanticAnalysis
Description	Retrieves the keywords exploiting frequency lists corpus and WordNet synsets relations
Input parameters	RESOURCE:InputResource – The text document from which keywords have to be extracted UINT16:MaxKWNumber – The maximum number of keyword requested
Output parameters	STRING:Keywords – A string containing keywords separated by return + newline chars

Within MPEG-7 the following tools are relevant for the meta data extracted from text and documents:

Tool	*Functionality*
Language Identification	Tools for identifying the language of a textual description or of the AV content itself. MPEG-7 uses the XML defined xml:lang attribute to identify the language used to write a textual description.
Text Annotation	Tools for representing unstructured and structured textual annotations. Unstructured annotations (i.e. with free text) are represented using the FreeTextAnnotation datatype. Annotations that are structured in terms of answering the questions "Who? What? Where? How? Why?" are represented using the StructuredAnnotation datatype. Annotations structured as a set of keywords are representation using the KeywordAnnotation datatype. Finally, annotations structured by syntactic dependency relations-for example, the relation between a verb phrase and the subject-are represented using the DependencyStructure datatype.

12.2 Wordnet

(http://wordnet.princeton.edu/)
WordNet® is an electronic lexical database in which nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept.
WordNet is distributed under an “AS-IS” license and can be used in commercial applications without restrictions.
For non-English languages, ELDA (http://www.elda.org/) distributes comparable databases under research and commercial licenses. A license for research will be provided by DSI.

12.3 TreeTagger

(http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html)
The TreeTagger is a tool for annotating text with part-of-speech and lemma information which has been developed within the TC project at the Institute for Computational Linguistics of the University of Stuttgart.
The TreeTagger has been successfully used to tag German, English, French, Italian, Greek and old French texts and is easily adaptable to other languages if a lexicon and a manually tagged training corpus are available.
TreeTagger is freely available for research, education and evaluation. It’s not redistributable.
Commercial licenses have to be defined contacting the author the Institut fuer maschinelle Sprachverarbeitung, Universitaet Stuttgart.

12.4 WordNet Domains

(http://wndomains.itc.it/)
Semantic Domains provide a natural way to establish semantic relations among word senses, which can be profitably used for Computational Linguistics. They are areas of human discussion, such as POLITICS,

ECONOMY, SPORT, which exhibit their own terminology and lexical coherence. Domains have been used both in Linguistics (i.e. Semantic Fields) and in Lexicography (i.e. Subject Field Codes) to mark technical usages of words. Semantic Domains can also be used to describe texts according to general subjects (topics) characterized by domain specific lexicon.
WordNet Domains has augmented the Princeton English WordNet with some Domain Labels. Synsets have been annotated with at least one domain label, selected from a set of about two hundred labels hierarchically organized. Information brought by domains is complementary to what is already in Wordnet. A domain may include synsets of different syntactic categories and from different Wordnet sub-hierarchies. Domains may group senses of the same word into homogeneous clusters, with the side effect of reducing word polysemy in Wordnet.

13 Descriptor extractor as fingerprint for Audio files (EPFL)

Module/Tool Profile Tools and Algorithms for Audio Adaptation		
Responsible Name	Mattavelli	
Responsible Partner	EPFL	
Status (proposed/approved)	Proposed	
Implemented/not implemented	Implemented	
Status of the implementation	Plug-In	
Executable or Library/module (Support)	Library	
Single Thread or Multithread	Single	
Language of Development	C++	
Platforms supported	Win32	
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/newrepos/Framework/src/descriptors/audio	
Reference to the AXFW location of the demonstrator executable tool for internal download	https://cvs.axmedis.org/newrepos/Framework/bin/descriptors/audio	
Reference to the AXFW location of the demonstrator executable tool for public download	--	
Address for accessing to WebServices if any, add accession information (user aNd Passwd) if any	--	
Test cases (present/absent)	Absent	
Test cases location	--	
Usage of the AXMEDIS configuration manager (yes/no)	No	
Usage of the AXMEDIS Error Manager (yes/no)	No	
Major Problems not solved	-- --	
Major pending requirements	-- --	
Interfaces API with other tools, named as	Name of the communicating tools References to other major components needed	Communication model and format (protected or not, etc.)
Formats Used	Shared with	format name or reference to a section

Protocol Used	Shared with	Protocol name or reference to a section
Used Database name		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
LIBSNDFILE		LGPL

13.1.1 General Description of the Module

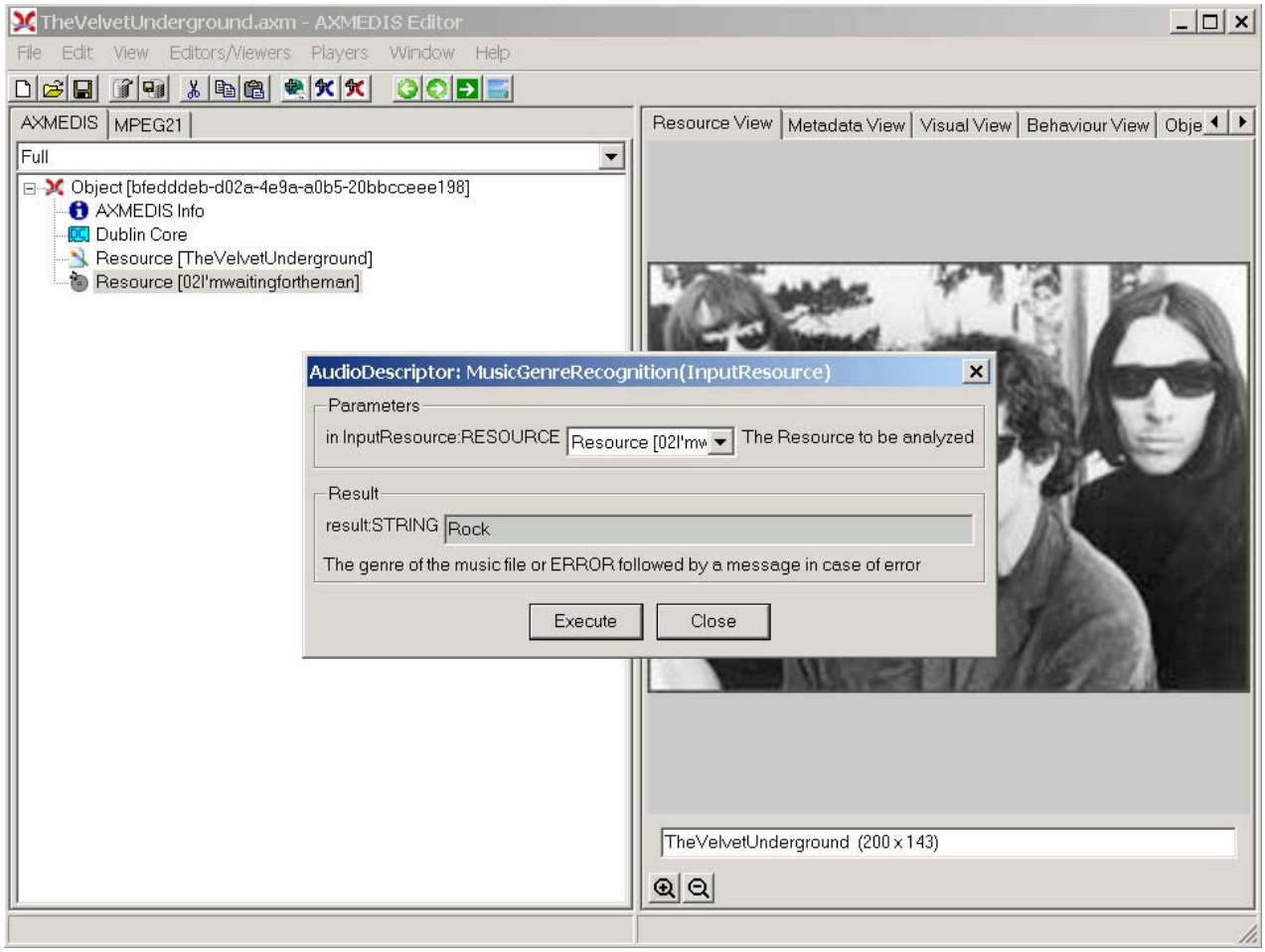
With electronic music distribution (EMD), music catalogues have become huge. The biggest online services now propose around 2 million tracks while personal users have the possibility to carry thousand of songs on their portable music player. In fact, the amount of digital music is now urging for reliable and fast tools for content analysis and description, to be used for searches, content queries and interactive access. In the context of the AXMEDIS project, a number of algorithms for audio content analysis and description have been developed and implemented to ease audio content retrieval and browsing into collections.

13.1.2 User interface description

The audio descriptors extraction functionalities are to be used as plug-ins through the AXCP interface. The interface of AXCP plug-ins maps exactly the formal description of the function and allows entering textually all parameters of the function. Moreover, it displays a brief description of the meaning of the parameters of the function to ease their use. The result of the adaptation is displayed as a textual message in the *Result* box of the interface.

The adaptation is launched by clicking the *Execute* button and the window can be closed with the *Close* button once the adaptation has been performed.

The following figure shows the user interface of the music genre recognizer function. Please refer to section 15.4.10 for the formal description of the music genre recognizer function and to understand how the user interface reflects this formal description.



13.1.3 Technical and Installation information

The audio descriptors extraction functionalities are to be used as plug-ins through the AXCP interface. The plug-in simply consists of a DLL and an XML file describing the functionalities of the DLL. Both the DLL and the XML description should be installed in the plug-in directory of the AXCP compliant tool using the plug-in.

References to other major components needed	The following DLLs need to be loaded for the correct execution of the audio adaptation tool. The simplest solution is to copy them into the directory of the AXCP compliant tool using the audio adaptation functionalities: • libsndfile.dll
Problems not solved	• NONE
Configuration and execution context	--

13.1.4 Draft User Manual

13.1.4.1 Low-Level Descriptors

The Low-Level Descriptors extractor allows extracting morphological descriptors of the audio signal:

- **AudioWaveform**: describes the audio waveform envelope, typically for display purpose.
- **AudioPower**: describes the temporally smoothed instantaneous power.
- **AudioSpectrumEnvelope**: describes the spectrum of the audio according to a logarithmic frequency scale.
- **AudioSpectrumCentroid**: describes the center of gravity of the log-frequency power spectrum.
- **AudioSpectrumSpread**: describes the second moment of the log-frequency power spectrum.
- **AudioSpectrumFlatness**: describes the flatness properties of the spectrum of an audio signal within a given number of frequency bands.
- **Mel Frequency Energies:** energy on the Mel scale which is a perceptually motivated scale of pitches.
- **MFCCs**: Mel-Frequency Cepstral Coefficients, e.g. spectral shape descriptors.
- **ZCR**: number of time domain zero crossing of the signal.

13.1.4.2 Audio Files Segmentation

The Speech / Noise / Music discriminator allows segmenting the audio stream into three kind of semantically coherent segments:

- Speech segment: speech segments are defined as regions of the audio file in which spoken content is dominant.
- Music segment: music segments are defined as regions of the audio file in which music content is dominant.
- Noise segment: noise segments are defined as regions of the audio file in which noise is dominant; noise is loosely defined as audio content which is not speech, music nor silence.

13.1.4.3 Music Genre Recognizer

The Music Genre recognizer allows characterizing music segments in terms of music genres. The provided model classifies music in the following genres:

- Classical
- Jazz
- Electronic
- Rap
- Rock

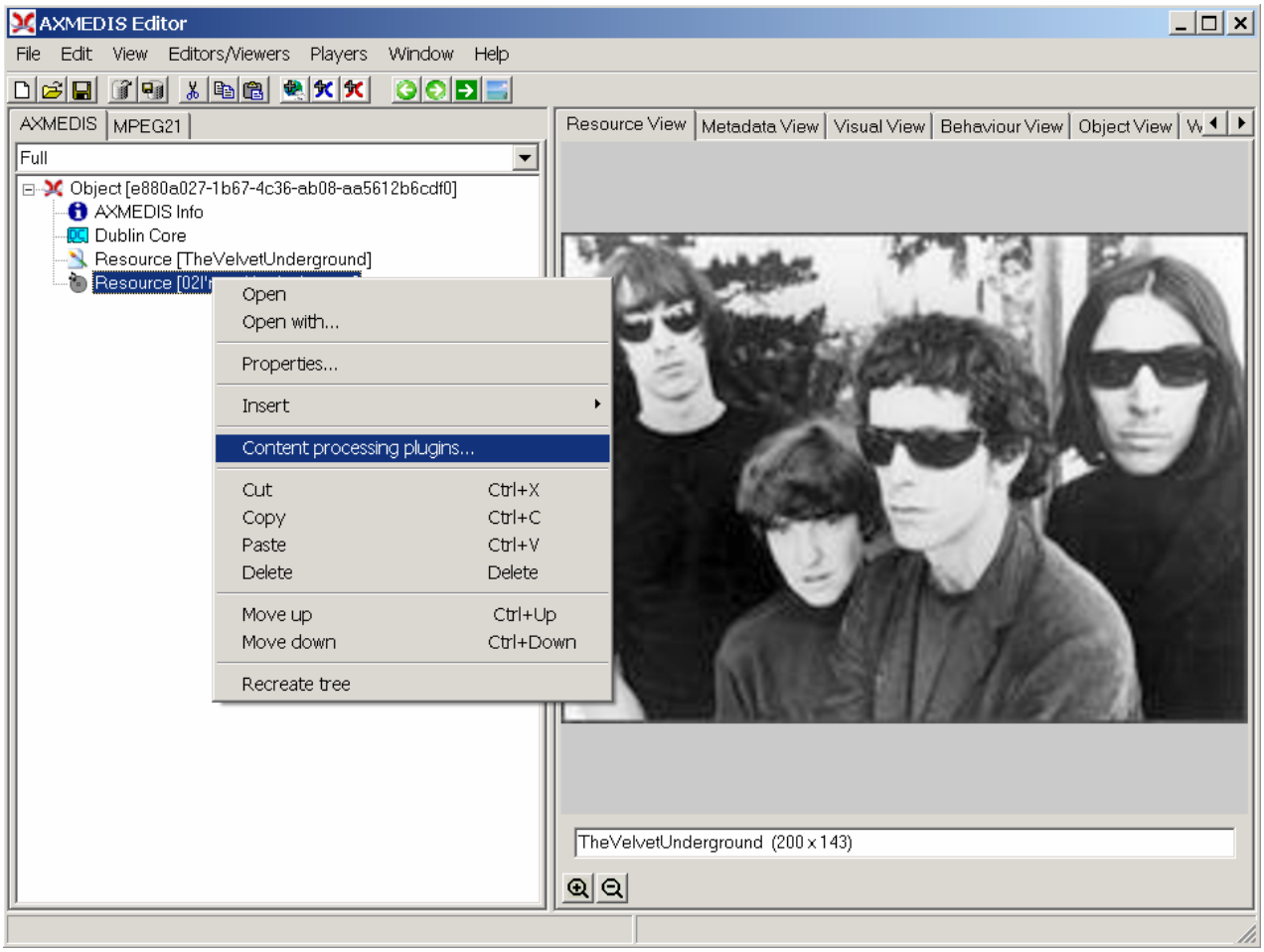
13.1.4.4 Rhythm Characterization

The rhythm characterization tool allows extracting tempo and meter from audio files.

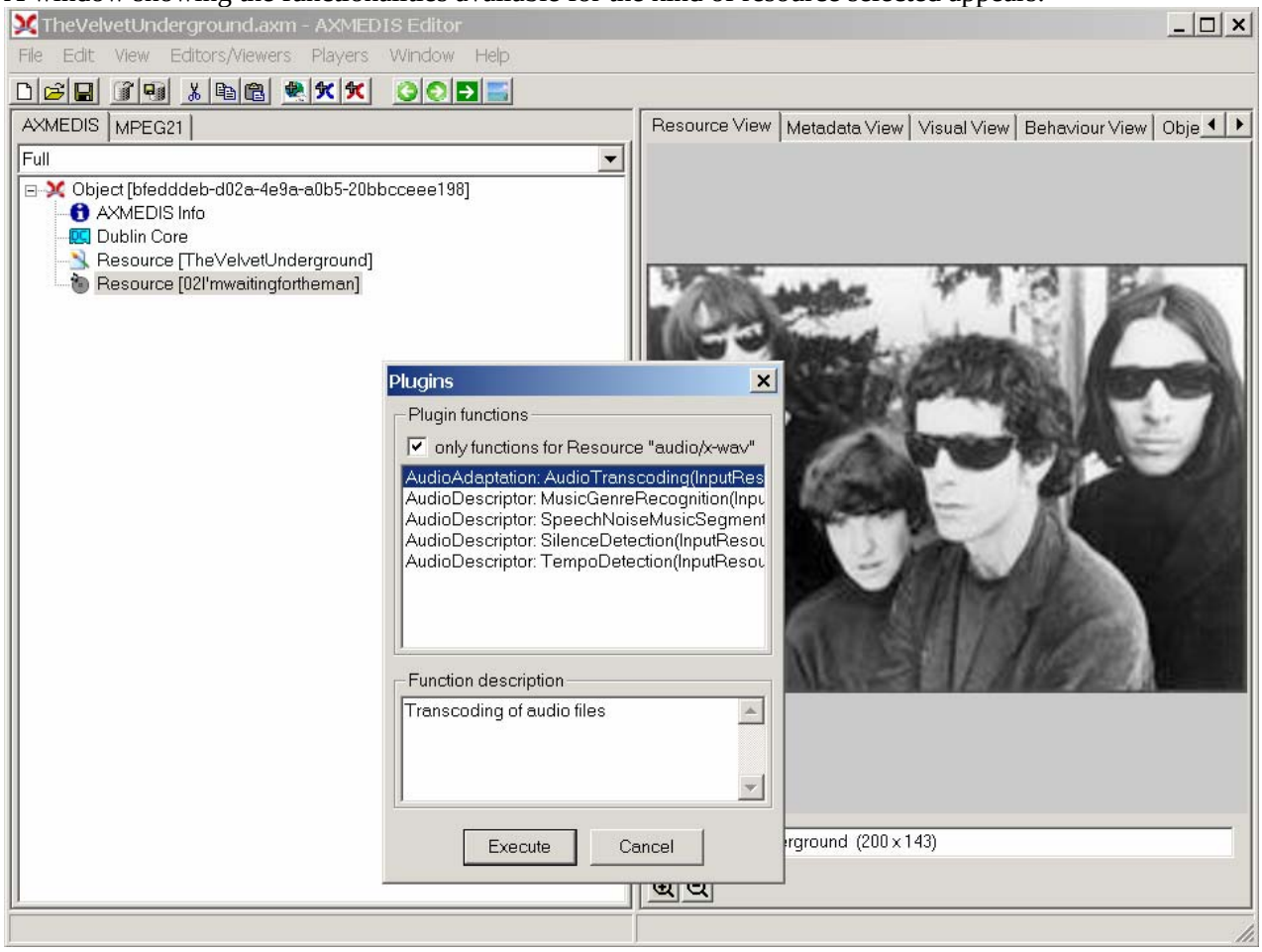
13.1.5 Examples of usage

Here’s an example on how to use the music genre recognition function as a plug-in with for the AXMEDIS editor.

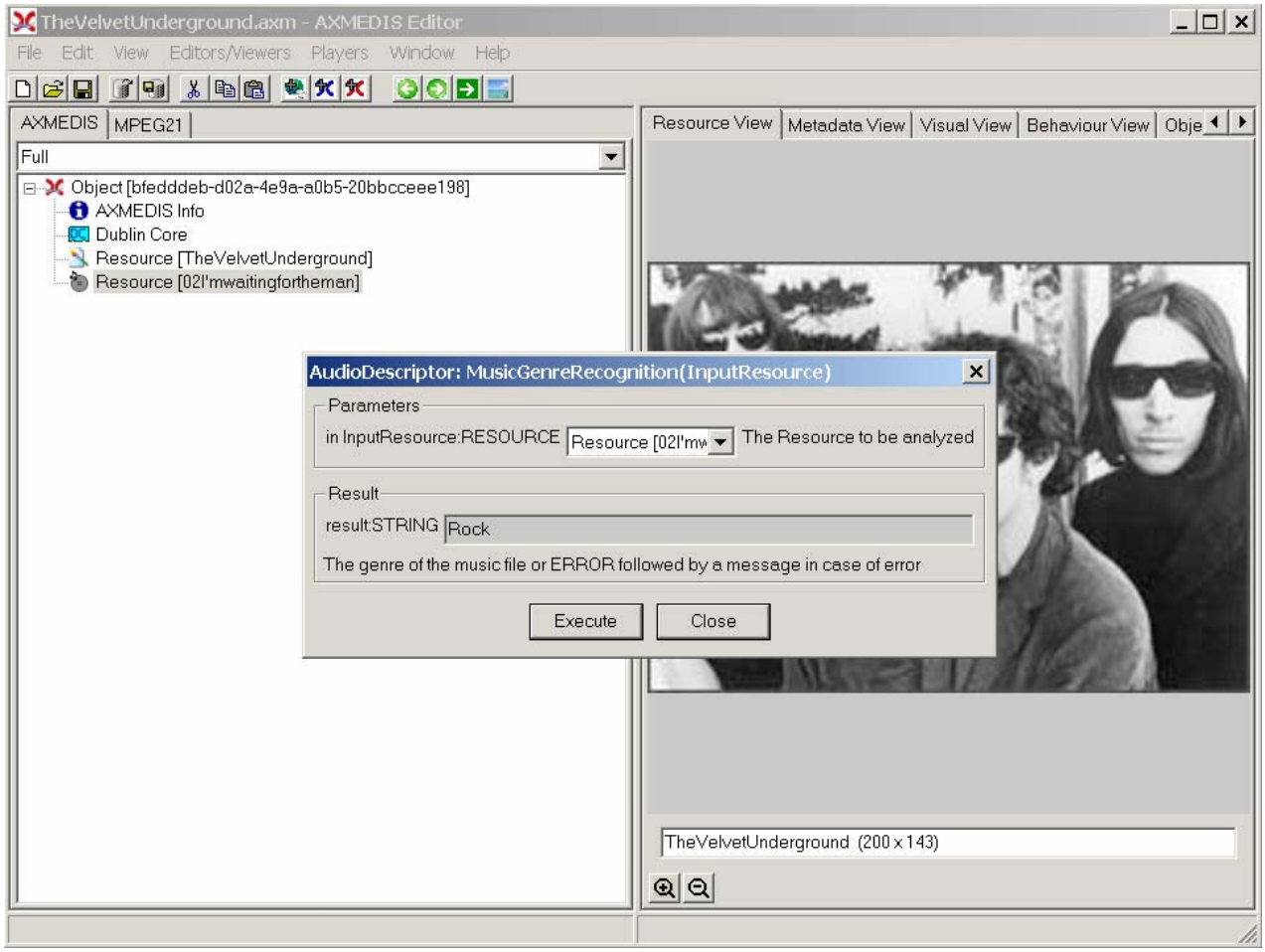
The plug-in must be applied on an audio resource of an AXMEDIS object. The adaptation plug-in is called by right-clicking on the interesting resource and selecting the ‘Plugin…’ command:



A window showing the functionalities available for the kind of resource selected appears:



The Music Genre recognizer is called by selecting the **AudioDescriptor: MusicGenreRecognition** function. A new window appears showing the interface to the music genre recognizer (see below). In the current implementation, the result of the recognition is displayed in the **Result** part of the graphical interface. In future implementation, an MPEG-7 compliant description of the audio segment along with its genre label will be produced and saved in the AXMEDIS object to allow for intelligent retrieval of audio files.



13.1.6 Integration and compilation issues

The tool has been compiled and tested successfully on Win32 platform. It should be ported easily on Linux and MacOsX platforms though it has not been tested yet.

13.2 Low-Level Audio Descriptors

By Low-Level Audio Descriptors (LLDs), we refer here to simple and low complexity descriptors that can be extracted automatically from the audio data in a systematic way and that represent audio signals in an objective manner.

Such descriptors are purely morphologic i.e. they do not carry any information on the actual meaning of the source or in other words they do not have a direct mapping to a high-level human percept. On the contrary, LLDs refer to the inner structural elements of the signal such as energies in some specific frequency bands or main spectral components etc.

Extraction of LLDs is crucial however since their combination (with automated learning techniques forexample) allows the building of higher-level descriptors i.e. descriptors which actually have a semantic or syntactic meaning for human users.

In the context of MPEG-7 [1], a standardization initiative of the Motion Picture Expert Group meant to describe multimedia content, a number of LLDs have been described. The following MPEG-7 LLDs have notably been implemented in the context of AXMEDIS:

- **AudioWaveform**: describes the audio waveform envelope, typically for display purpose.
- **AudioPower**: describes the temporally smoothed instantaneous power.
- **AudioSpectrumEnvelope**: describes the spectrum of the audio according to a logarithmic frequency scale.

- **AudioSpectrumCentroid**: describes the center of gravity of the log-frequency power spectrum.
- **AudioSpectrumSpread**: describes the second moment of the log-frequency power spectrum.
- **AudioSpectrumFlatness**: describes the flatness properties of the spectrum of an audio signal within a given number of frequency bands.

MPEG-7 also proposes a low-dimensional description of a spectrum obtained by projection on a reduced rank basis obtained by singular value decomposition. Mel-Frequency Cepstral Coefficients (MFCCs) describe the spectral shape of an audio signal in a similar way and have been widely used in the contexts of speech recognition [2] and music information retrieval [3]. Recent experiments [4] seem to demonstrate that MFCCs yield similar or even better results than MPEG-7 spectrum projection in a variety of applications. Consequently, we choose to implement MFCCs rather than MPEG-7 spectrum projections since the former are simpler to extract than the latter. As a side product of MFCCs extraction, the spectrum according to the Mel scale is evaluated. The Mel scale is a perceptually relevant scale of pitches, which is slightly different from the logarithmic scale used in the MPEG-7 **AudioSpectrumEnvelope** descriptor.

[1] MPEG-7, “Information Technology – Multimedia Content Description Interface – Part 4: Audio”, ISO/IEC JTC 1/SC29, ISO/IEC FDIS 15938-4:2002, 2002.
[2] L. Rabiner, B.H. Juang, “Fundamentals of speech recognition”, Englewood Cliffs, NJ, Prentice-Hall, 1993.
[3] B. Logan, “Mel Frequency Cepstral Coefficients for Music Modeling”, in International Symposium on Music Information Retrieval, 2000.
[4] H. Kim, T. Sikora, “Comparison of MPEG-7 Audio Spectrum Projection Features and MFCC Applied to Speaker Recognition, Sound Classification and Audio Segmentation”, in Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Montreal, Canada, 2004.

13.3 Audio Files Segmentation

Audio segmentation consists in segmenting a continuous audio stream in terms of acoustically homogeneous regions (the definition of homogeneity of regions actually depends on the task considered). Segmentation plays an important role in the pre-processing stages of analysis systems since it allows using descriptors extraction algorithms dedicated to specific audio segments and consequently allows improving transcription accuracy. Moreover, segmentation is useful in itself for indexing and browsing audio documents. For example, it allows navigating efficiently in large multimedia documents such as recorded radio web cast or movies.

The semantic regions we want to identify at first reflect the basic physical structure of the audio file and are the following:

- Silence
- Spoken content
- Music content
- Other noises

The segmentation process is as follow. Firstly the signal is parameterized locally in terms LLDs. The LLDs considered here are MFCCs, which give a good description of timbre and the modulation of the spectrum envelope around 4 Hz and 2 Hz, which corresponds respectively to typical rates of human speech and music. Low order statistics of these LLDs are computed over 250 milliseconds windows to diminish their variability.

To each window is associated an estimation of the probability of belonging to each of the 4 considered classes. This probability is obtained by feeding the low order statistics of LLDs to a Support Vector Machine (SVM) previously trained in a supervised manner. SVMs are highly efficient classifiers based on the structure risk minimization inductive principle and non-linear projection into high-dimension feature spaces. For more details, please refer to some authoritative literature [1].

Once class conditional posterior probabilities estimated for each window, a segmentation of the file is obtained by using the Viterbi algorithm to find the best possible state sequence, which could have emitted this observation sequence, according to the maximum likelihood criterion. This algorithm is similar to a 4 state fully connected Hidden Markov Model [2] with state posterior probabilities being estimated with SVMs. The state transition probabilities are set manually to favor staying in the same class for a minimum duration. The initial probabilities are also set manually to make classes equally likely at the beginning of the stream.

[1] C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", in Data Mining and Knowledge Discovery, 2(2): 121-167, 1998.
[2] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", in Proc. of the IEEE, vol. 37, no. 2, pp. 257-86, 1989.

13.4 Music Genre Recognition

Musical genres are categories that have arisen through a complex interplay of cultures, artists and market forces to characterize similarities between musicians or compositions. Though they may represent a simplification of one artist's musical discourse, they are crucial descriptors of music content since they have been widely used for years to organize music catalogues, libraries and music stores.

At the same time, even if terms such as *jazz*, *rock* or *pop* are widely used, they often remain loosely defined so that the problem of automatic genre classification becomes a non-trivial task. A lot of researchers have focused their attention in the recent years on this classification problem (see [1] for a review) and an evaluation of music genres classification algorithms has even been conducted at the Music Information Retrieval Evaluation exchange 2005 (MIREX 2005: http://www.music-ir.org/mirex2005) on 2 databases of audio files (one composed of 1515 songs over 10 genres and the other of 1414 songs over 6 genres).

The music genre classification algorithm implemented in the AXMEDIS framework achieved 74.99% normalized classification accuracy on the MIREX 2005 datasets (14 algorithms were evaluated with accuracies between 77.98% and 51.83%). Here follows a brief overview of this algorithm. For a more complete presentation, please refer to [2].

Three different sets of LLDs characterizing audio content are considered to determine genre: timbral features (MFCCs), intensity features (notably log compressed energies in different frequency bands) and some rhythmic features (extracted from the periodicity function used to estimate tempo; see section 15.4). Low order statistics of these LLDs are computed over 1 second window to diminish their variability.

For each texture window, a local decision about the music genre is evaluated. The local decision is given by independent SVMs specialized on the different LLDs sets. These SVMs receive as input the information features of the texture window plus those of the surrounding windows to provide for contextual. A single decision is obtained for the considered music excerpt by averaging the outputs of each SVM over time.

[1] N. Scaringella, G. Zoia, D. Mlynek, "Automatic Genre Classification of Music Content: A Survey", in IEEE Signal Processing Magazine: Special Issue on Semantic Retrieval of Multimedia, vol. 23, no. 2, pp. 133-141, March, 2006.
[2] N. Scaringella, D. Mlynek, "A Mixture of Support Vector Machines for Audio Classification", Music Information Retrieval Evaluation exchange, 2005 [Online]. Avalaible: http://www.music-ir.org/evaluation/mirex-results/audio-genre/scaringella.pdf.

13.5 Rhythm Description

A precise definition of musical rhythm does not exist. Most authors converge on the idea of *temporal regularity*. As a matter of fact, the perceived regularity is distinctive of rhythm and distinguishes it from *non-rhythm*. Extracting rhythmic information from music signals allows retrieving rhythmically similar items and can facilitate synchronisation between audio signals or audio and video for example.

A review of automatic rhythm description systems may be found in [1]. These automatic systems may be oriented towards different applications dedicated to rhythm: tempo induction, beat tracking, meter induction, quantization of performed rhythm, or characterization of intentional timing deviations.

The simplest and probably most important descriptor of musical rhythm is certainly tempo. It is indeed correlated to the perceived *speed* of a song and consequently makes sense to any listener. Moreover, it seems also correlated with the perceived intensity of a song, which can be loosely defined as the subjective impression of energy that music titles convey (see [2]).

The tempo induction algorithm implemented in the AXMEDIS framework was proposed by Klapuri *et al.* in [3]. This algorithm won the tempo induction algorithm contest held as part of the International Symposium on Music Information Retrieval (ISMIR) in 2004 [4] with an overall accuracy of 76.15% on 3199 test samples (12 algorithms were evaluated with accuracies between 76.15% and 39.95%). We provide here a brief overview of this algorithm.

In a first step, a time frequency analysis of the audio signal is performed. In our implementation, this analysis is done using the **AudioSpectrumEnvelope** LLD (see section 15.2). From this representation, a measure of the degree of musical accentuation as a function of time is evaluated at 4 different frequency ranges. Periodicities of these musical accentuation functions are analyzed thanks to a bank of comb filter resonators and combined into a single periodicity function. A probabilistic model including prior knowledge of musical meter and taking into account the temporal dependencies between successive estimates is used in a final step to do successive estimation of tempo along time.

The same algorithm can be used to estimate musical bars length. The ratio of the bar length and the musical beats period (which is inversely proportional to the tempo) allows estimating if the meter of the piece is binary or ternary.

[1] F. Gouyon, S. Dixon, “A review of automatic rhythm description system”, in Computer Music Journal, vol. 29, pp. 34-54, 2005.
[2] A. Zils, F. Pachet, “Extracting automatically the perceived intensity of music titles”, in Proc. of the 6th Int. Conf. on Digital Audio Effects (DAFX-03), London, UK, September 8-11, 2003.
[3] A. Klapuri, A. Eronen, J. Astola, “Analysis of the Meter of Acoustic Musical Signals”, in IEEE Transactions on Speech and Audio Processing, Vol. 14(1), 2006.
[4] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, P. Cano, “An Experimental Comparison of Audio Tempo Induction Algorithms”, in IEEE Transactions on Speech and Audio Processing, Vol. 14(5), 2006.

14 Descriptor extractor as fingerprint for Video Files (FHGIGD)

Module/Tool Profile		
MPEG7		
Responsible Name	Martin Schmucker	
Responsible Partner	FHGIGD	
Status (proposed/approved)	Proposed	
Implemented/not implemented	Implemented	
Status of the implementation	100%	
Executable or Library/module (Support)	Library	
Single Thread or Multithread	Single	
Language of Development	C++	
Platforms supported	Win32	
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/newrepos/Framework/source/descriptor/video https://cvs.axmedis.org/newrepos/Framework/project/descriptor/video https://cvs.axmedis.org/newrepos/Framework/include/descriptor/video	
Reference to the AXFW location of the demonstrator executable tool for internal download	https://cvs.axmedis.org/newrepos/Framework/bin/fingerprint/metadata	
Reference to the AXFW location of the demonstrator executable tool for public download	http://www.axmedis.org	
Address for accessing to WebServices if any, add accession information (user and Passwd) if any	not included yet	
Test cases (present/absent)	Absent	
Test cases location	-	
Usage of the AXMEDIS configuration manager (yes/no)	No	
Usage of the AXMEDIS Error Manager (yes/no)	No	
Major Problems not solved	-- --	
Major pending requirements	-- --	
Interfaces API with other tools, named as	Name of the communicating tools References to other major components needed	Communication model and format (protected or not, etc.)
Formats Used	Shared with	format name or reference to a section
Protocol Used	Shared with	Protocol name or reference to a section

Used Database name		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not

14.1.1 General Description of the Module

Within the AXMEDIS project no research on the content descriptors for video is performed. Instead, the extensibility of the AXMEDIS plug-in interface is exploited and validated by integrating existing state-of-the-art algorithms.

The MPEG-7 eXperimental Model (XM) is the MPEG-7 Reference Software and is available for downloading at http://www.lis.e-technik.tu-muenchen.de/research/bv/topics/mmdb/mpeg7.html and via CVS.[1] The MPEG-7 XM Reference software includes state-of-the-art content description algorithms for audio-visual content.

Some of the available functionality was selected based on the identified user requirements. These functionalities are:

- Homogeneous Texture Descriptor
- Dominant Color Descriptor
- GoF/GoP Descriptor
- Color Structure Descriptor

A detailed description is given below in the subsection on the formal description of the algorithm.

14.1.2 Module Design in terms of Classes

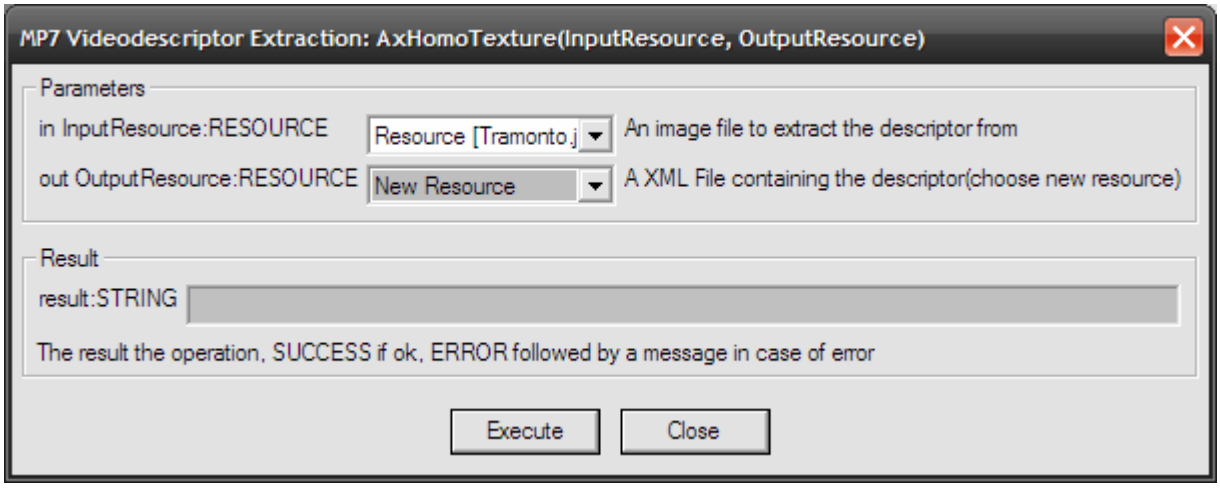
The functionality is integrated in separate plug-ins.

14.1.3 User interface description

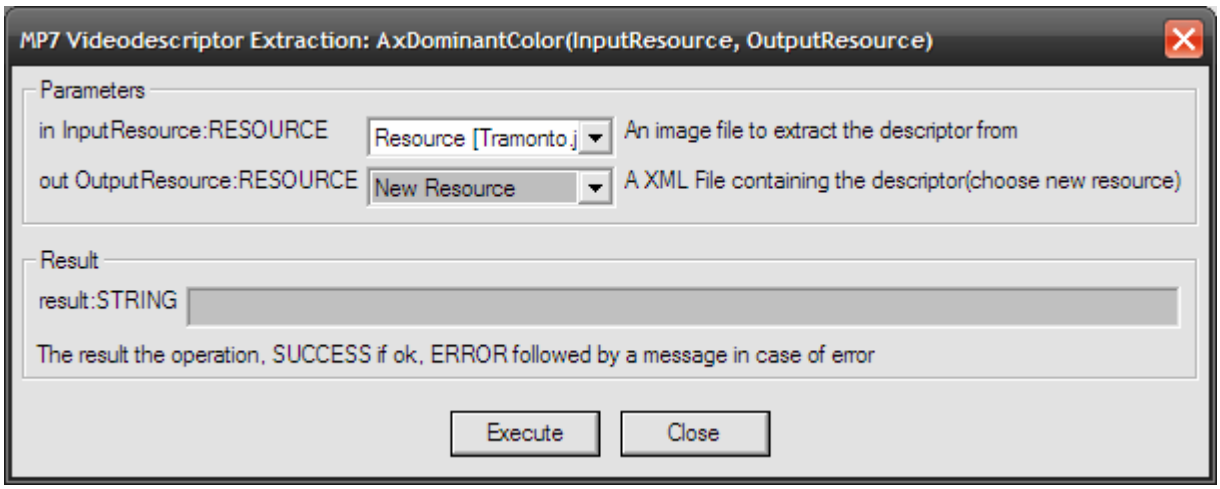
Usage of the developed plug-ins depends on the AXMEDIS program that utilizes the available functionality. As shown in the next figures – which are examples when the plug-ins are used in the AXEditor – several parameters can be set.

Homogeneous Texture Descriptor

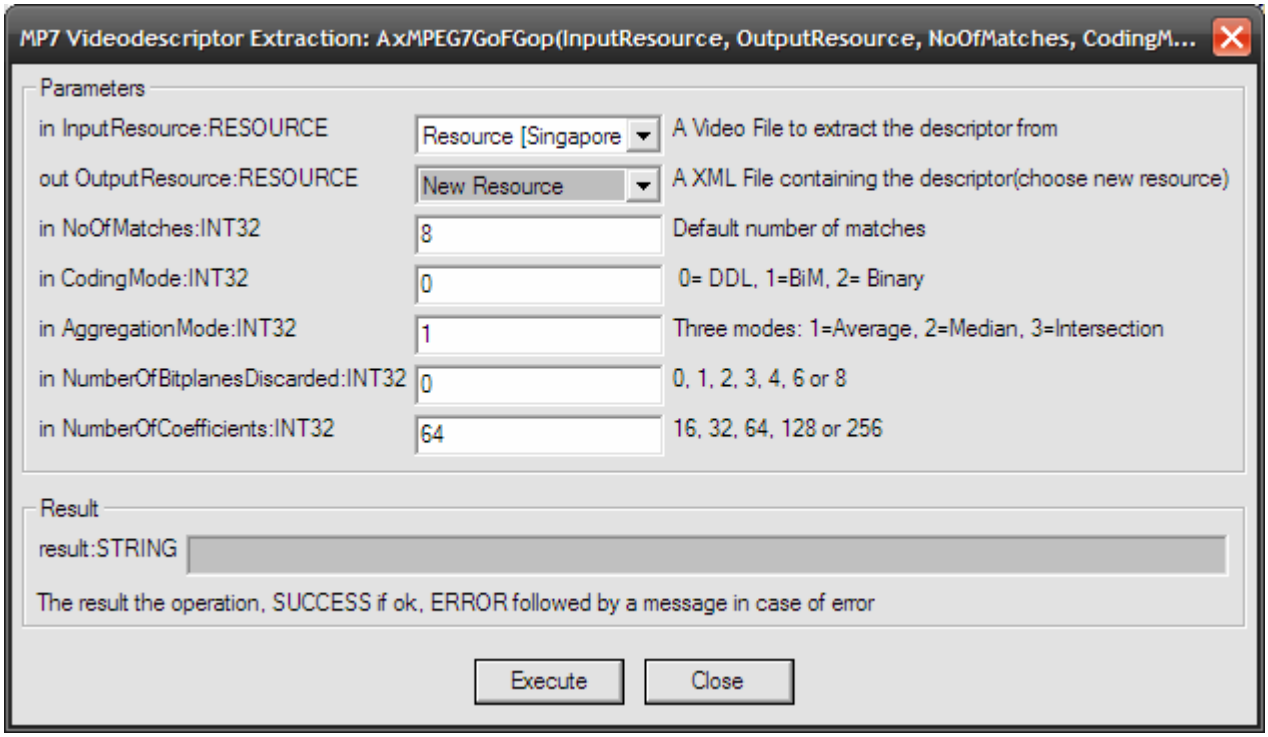
[1] Access to the CVS is only possible through registration at the national standardization body.



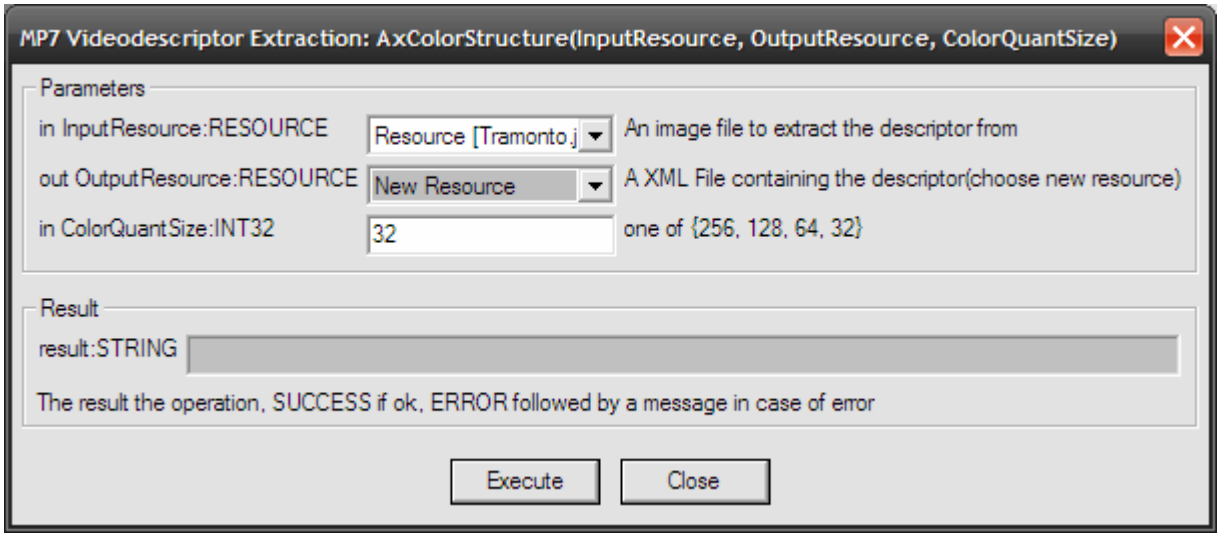
Dominant Color Descriptor



GoF/GoP Descriptor



Color Structure Descriptor



14.1.4 Technical and Installation information

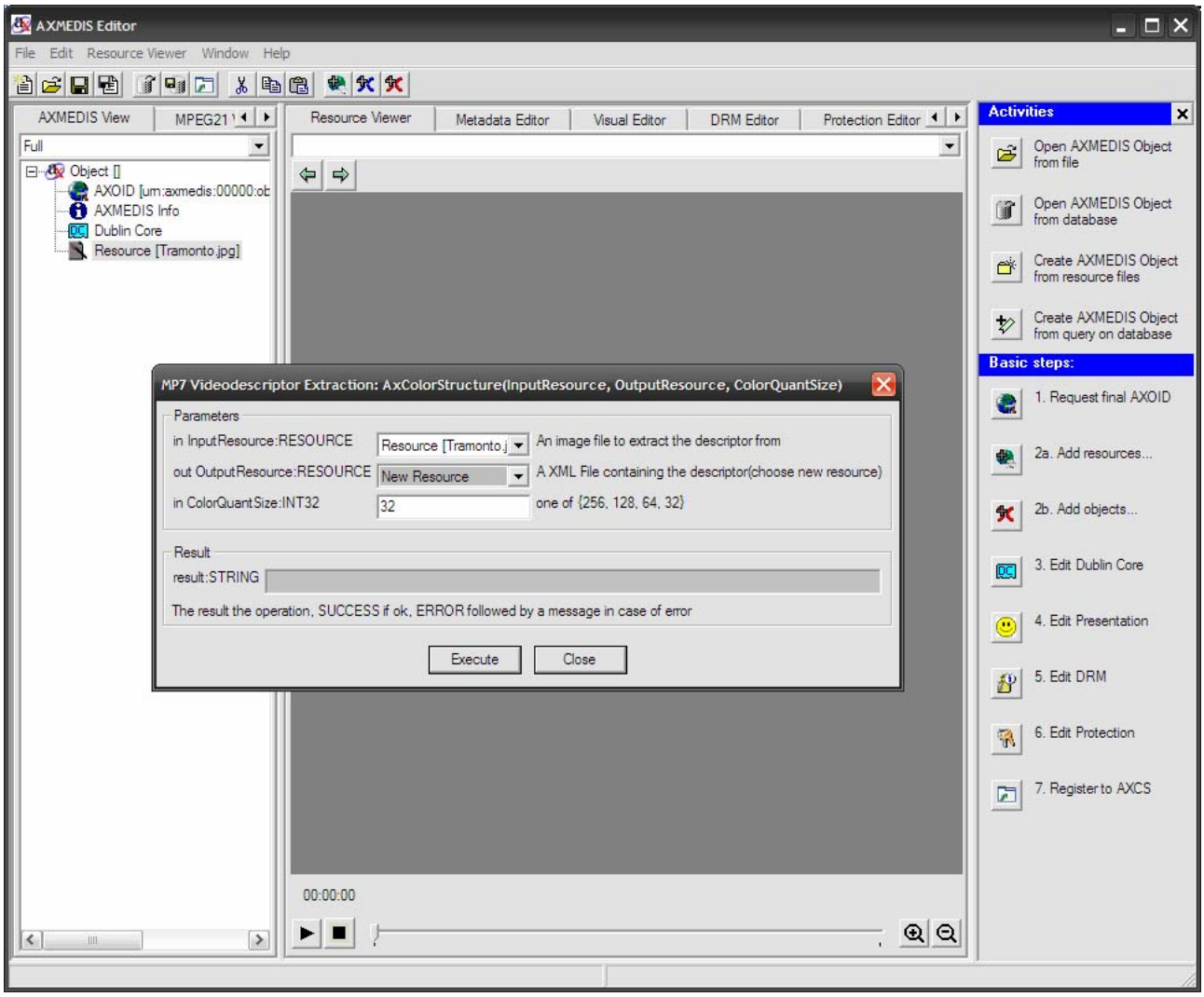
References to other major components needed	AXMP7IMagick.dll, AXMP7MagickJBIG.dll, AXMP7MagickJPEG.dll, AXMP7MagickPNG.dll, AXMP7MagickTIFF.dll, AXMP7Magickttf.dll, AXMP7MagickZIP.dll, cv.dll, mpeg2decode.exe, xerces-c_1_6_0.dll and XMWinExe.exe
Problems not solved	for GoF/GoP some videos could not be used as input due to the limitations of the XM-Decoder
Configuration and execution context	Plug-ins are installed by copying the library and the library description in the corresponding plug-in directory.

14.1.5 Draft User Manual and Examples of Usage

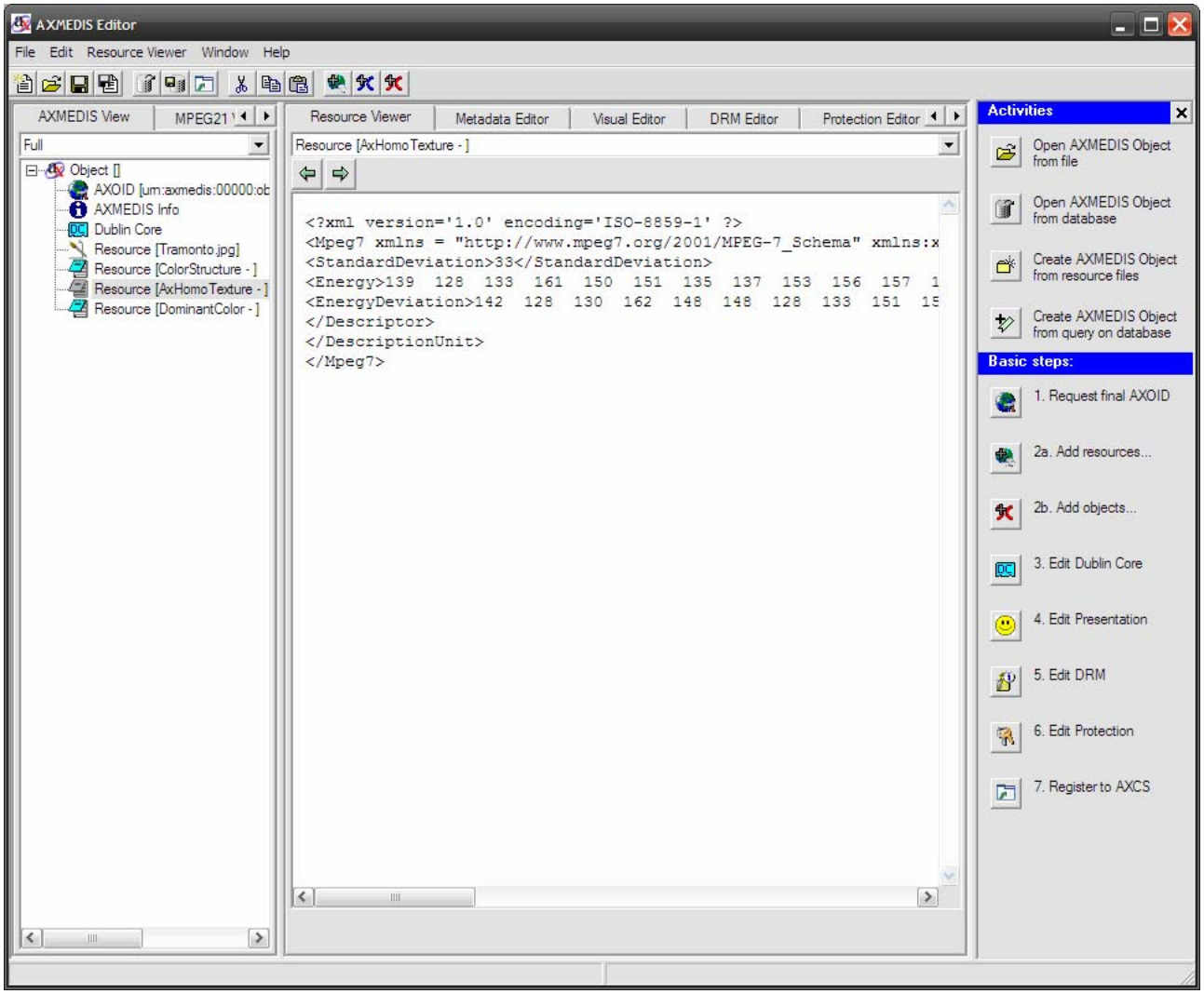
The metadata-plug-in can be applied to different MIME types. The GoF/GoP descriptor can be applied to MPEG video. The other three descriptors can be applied to image resources, provided this was declared in the mime type attribute of the resource.

Before using the metadata-plug-in an AXMEDIS object containing a video resource has to be opened. Alternatively, a new AXMEDIS oject can be created and a corresponding resource has to be added.

For the resource the "Content Processing Plug-in…" command has to be selected and the desired MP7 descriptor has to be chosen.



The result of each MP7 descripto is an XML-resource. The following example shows the result of the Homogeneous Texture Descriptor:



More detailed information is available in the plug-in description.

14.1.6 Formal description of algorithm

14.1.6.1 Homogeneous Texture Descriptor

Document	MPEG-7 Visual XM and CD, see Homogeneous Texture Descriptor
Name	yanglim Choi, M/M Lab, Samsung Advanced Institute of Technology
E-Mail	yanglimc@samsung.com
Input	Images/Regions(JPEG,BMP,etc)
Summary	This component generates a texture descriptor for a homogeneously textured Image/Region for Search and Retrieval using texture features. The descriptor components are the average and the standard deviation of the image together with the energies and the energy deviations of Gabor filtered reponse of each (predefined) frequency channels.
Strong Points	Statistically very precise description of homogeneously textured region.
Limitations	Relatively large in size (32 components for the basic layer and 62 components for the enhanced layer).
Known Problems	NONE
Parameters	Plugin implementation set to standard values.

14.1.6.2 Dominant Color Descriptor

Document	MPEG-7 Visual XM and CD, see Dominant Color Descriptor
Name	Jungmin Song, Heon Jun Kim, Leszek Cieplinski, Prof. Manjunath
	EMail jmsong73@mail.lgcit.com, hjk@lge.co.kr, Leszek.Cieplinski@vil.ite.mee.com, manj@ece.ucsb.edu
Contact	Jungmin Song (general), Leszek Cieplinski(color variance), Prof. Manjunath(dominant color extraction and search algorithm)
Type	application
External Libraries	none
Related Ds/DSs	Color Space Descriptor, Color Quantization Descriptor, Related DS's are not defined fully yet.
Used Ds/DSs	Color Space Descriptor, Color Quantization Descriptor
Input	Images
Extraction	Yes
Client Appl	Search, Retrieval, Browsing
Summary	The dominant color descriptor is useful for image and video retrieval. It targets content-based retrieval for color, either for the whole image or for any arbitrary shaped region (rectangular or irregular). It is a very compact descriptor, requiring less than 6-8 colors per region. Since colors are not pre-quantized as in the histogram type color descriptors, the representation is more accurate. To accomplish high accuracy in retrieval, Spatial Coherency and/or Color Variance can be utilized. It is intended for applications that use object based representations (objects or regions in an image).
Limitations	The maximum allowed number of dominant colors is 8.
Known Problems	None.
Parameters	color space and color quantization parameters

14.1.6.3 GoF/GoP Color Descriptor

Document	MPEG-7 Visual XM and CD, see GoF/GoP Color
Name	Santhana Krishnamachari, Mufit Ferman
Contact	santhana.krishnamachari@philips.com
Type	Application
External Libraries	None
Related Ds/DSs	Scalable Color, Video Segment DS, Collection DS
Used Ds/DSs	Scalable Color
Input	Video, Image
Extraction	Yes
Client Appl	Search & Retrieval
Summary	This library implements the GoF/GoP Color descriptor which is used to describe the color characteristics of a collection of video frames (and a collection of images). It consists of one primary and four secondary attributes. Since the feature vector is short, a simple absolute distance or squared distance criterion can be used for matching.
Strong Points	None.
Limitations	Use mean or median aggregation for matching
Known Problems	None.
Parameters	None.

14.1.6.4 Color Structure Descriptor

Document	MPEG-7 Visual XM and CD, see Color Structure Descriptor
Name	Jim Errico, Sharp Labs of America; Dean Messing, Sharp Labs of America
E-Mail	jerrico@sharplabs.com; deanm@sharplabs.com
Input	Images
Summary	This component is the implementation of the extraction and search functionality for the ColorStructure Descriptor.

Strong Points	NA
Limitations	NA
Known Problems	NA
Parameters	ColorQuantSize : one of {256, 128, 64, 32}

15 Descriptors Formats (FHGIGD)

The descriptors format definition (as well as the fingerprints format definition) is based on the corresponding MPEG standardization. As meta-data is standardized in MPEG-7, the descriptors (as well as the fingerprint) format definitions correspond with the MPEG-7 descriptors. This section briefly describes the corresponding where they are suitable. If they are not suitable, an explanation is given.

15.1 Language Descriptors defined in MPEG-7 (DIPITA):

MPEG-7 distinguishes between the language of the metadata and the language of the content. (However, MPEG-7 has a strong focus on audio-visual content as described below). The xml:lang attribute must be used in the first case-for example, to specify the language in which a textual annotation is written-and the built-in XML Schema language datatype in the second-for example, e.g. to specify the language of an audio track.

```
<!-- ################################################ -->
<!-- Definition of Language Datatype                  -->
<!-- ################################################ -->

<!--- %%% The datatype is already defined in XML      -->
<!--- <simpleType name="language" base="string"/>  -->


<!-- ################################################ -->
<!--   Definition of Classification DS               -->
<!-- ################################################ -->
<complexType name="ClassificationType">
   <complexContent>
      <extension base="mpeg7:DSType">
         <sequence>
            <element name="Form" type="mpeg7:ControlledTermUseType" minOccurs="0"/>
            <element       name="Genre"       type="mpeg7:GenreType"       minOccurs="0"
maxOccurs="unbounded"/>
            <element name="Subject" type="mpeg7:TextAnnotationType" minOccurs="0"/>
            <element  name="Purpose"  type="mpeg7:ControlledTermUseType"  minOccurs="0"
maxOccurs="unbounded"/>
            <element  name="Language"  type="mpeg7:ExtendedLanguageType"  minOccurs="0"
maxOccurs="unbounded"/>
            <element      name="SubtitleLanguage"      type="mpeg7:ExtendedLanguageType"
minOccurs="0" maxOccurs="unbounded"/>
            <element    name="ClosedCaptionLanguage"    type="mpeg7:ExtendedLanguageType"
minOccurs="0" maxOccurs="unbounded"/>
            <element name="SignLanguage" minOccurs="0">
               <complexType>
                  <attribute name="primary" type="boolean" use="optional"/>
                  <attribute name="translation" type="boolean" use="optional"/>
               </complexType>
            </element>
            <element name="Release">
               <complexType>
                  <sequence>
                     <element   name="Country"   type="mpeg7:countryCode"   minOccurs="0"
maxOccurs="unbounded"/>
                  </sequence>
                  <attribute name="date" type="mpeg7:timePointType" use="optional"/>
               </complexType>
            </element>
            <element name="Target">
               <complexType>
                  <sequence>
                     <element      name="Market"      type="mpeg7:ControlledTermUseType"
minOccurs="0" maxOccurs="unbounded"/>
                     <element name="Age" minOccurs="0">
                        <complexType>
```

```
                        <attribute         name="min"         type="nonNegativeInteger"
use="optional"/>
                        <attribute         name="max"         type="nonNegativeInteger"
use="optional"/>
                     </complexType>
                  </element>
                  <element   name="Country"   type="mpeg7:countryCode"   minOccurs="0"
maxOccurs="unbounded"/>
               </sequence>
            </complexType>
         </element>
         <element      name="ParentalGuidance"      type="mpeg7:ParentalGuidanceType"
minOccurs="0" maxOccurs="unbounded"/>
         <element   name="MediaReview"   type="mpeg7:MediaReviewType"   minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
     </extension>
  </complexContent>
</complexType>


<!-- ################################################    -->
<!--   Definition of ExtendedLanguage Datatype           -->
<!-- ################################################    -->
<complexType name="ExtendedLanguageType">
   <simpleContent>
      <extension base="language">
         <attribute name="type" use="optional" default="main">
            <simpleType>
               <restriction base="string">
                  <enumeration value="main"/>
                  <enumeration value="original"/>
                  <enumeration value="other"/>
                  <enumeration value="backgroundOriginal"/>
               </restriction>
            </simpleType>
         </attribute>
      </extension>
   </simpleContent>
</complexType>
```

Keywords Descriptors defined in MPEG-7:

```
<!-- ############################################################  -->
   <!-- TextAnnotation Datatype                                      -->
   <!-- ############################################################  -->
   <complexType name="TextAnnotationType">
      <choice maxOccurs="unbounded">
         <element name="FreeTextAnnotation" type="mpeg7:TextualType"/>
         <element name="StructuredAnnotation" type="mpeg7:StructuredAnnotationType"/>
         <element name="DependencyStructure" type="mpeg7:DependencyStructureType"/>
         <element name="KeywordAnnotation" type="mpeg7:KeywordAnnotationType"/>
      </choice>
   </complexType>
   <!-- ########################################################## -->
   <!-- Definition of StructuredAnnotation Datatype           -->
   <!-- ########################################################## -->
   <complexType name="StructuredAnnotationType">
      <sequence>
         <element   name="Who"          type="mpeg7:TermUseType"          minOccurs="0"
maxOccurs="unbounded"/>
         <element   name="WhatObject"       type="mpeg7:TermUseType"       minOccurs="0"
maxOccurs="unbounded"/>
         <element   name="WhatAction"       type="mpeg7:TermUseType"       minOccurs="0"
maxOccurs="unbounded"/>
         <element   name="Where"         type="mpeg7:TermUseType"         minOccurs="0"
maxOccurs="unbounded"/>
```

```
        <element        name="When"        type="mpeg7:TermUseType"        minOccurs="0"
maxOccurs="unbounded"/>
        <element        name="Why"        type="mpeg7:TermUseType"        minOccurs="0"
maxOccurs="unbounded"/>
        <element        name="How"        type="mpeg7:TermUseType"        minOccurs="0"
maxOccurs="unbounded"/>
      </sequence>
      <attribute ref="xml:lang" use="optional"/>
   </complexType>
   <!-- ########################################################## -->
   <!-- Definition of KeywordAnnotation Datatype              -->
   <!-- ########################################################## -->
   <complexType name="KeywordAnnotationType">
      <sequence maxOccurs="unbounded">
         <element name="Keyword">
            <complexType>
               <simpleContent>
                  <extension base="mpeg7:TextualType">
                     <attribute name="type" use="optional" default="main">
                        <simpleType>
                           <restriction base="string">
                              <enumeration value="main"/>
                              <enumeration value="secondary"/>
                              <enumeration value="other"/>
                           </restriction>
                        </simpleType>
                     </attribute>
                  </extension>
               </simpleContent>
            </complexType>
         </element>
      </sequence>
      <attribute ref="xml:lang" use="optional"/>
   </complexType>
   <!-- ########################################################## -->
   <!-- Definition of Dependency Structure Datatype           -->
   <!-- ########################################################## -->
   <complexType name="DependencyStructureType">
      <sequence>
         <element       name="Sentence"       type="mpeg7:DependencyStructurePhraseType"
maxOccurs="unbounded"/>
      </sequence>
      <attribute ref="xml:lang" use="optional"/>
      <attribute name="phonogrammicAlphabet" use="optional">
         <simpleType>
            <union>
               <simpleType>
                  <restriction base="NMTOKEN">
                     <enumeration value="Roman"/>
                     <enumeration value="Kana"/>
                     <enumeration value="Hangul"/>
                     <enumeration value="Pinyin"/>
                  </restriction>
               </simpleType>
               <simpleType>
                  <restriction base="mpeg7:termReferenceType"/>
               </simpleType>
            </union>
         </simpleType>
      </attribute>
   </complexType>
   <!-- ############################################################  -->
   <!-- Definition of DependencyStructurePhraseType datatype        -->
   <!-- ############################################################  -->
   <complexType name="DependencyStructurePhraseType">
      <choice maxOccurs="unbounded">
         <element name="Quotation" type="mpeg7:DependencyStructurePhraseType"/>
         <element name="Phrase" type="mpeg7:DependencyStructurePhraseType"/>
```

```
         <element name="Head">
            <complexType>
               <simpleContent>
                  <extension base="string">
                     <attribute name="terms" use="optional">
                        <simpleType>
                           <list itemType="mpeg7:termReferenceType"/>
                        </simpleType>
                     </attribute>
                     <attribute name="identifier" type="ID" use="optional"/>
                     <attribute name="equal" type="IDREF" use="optional"/>
                     <attribute name="type" use="optional">
                        <simpleType>
                           <union>
                              <simpleType>
                                 <list>
                                    <simpleType>
                                       <restriction base="NMTOKEN">
                                          <enumeration value="noun"/>
                                          <enumeration value="pronoun"/>
                                          <enumeration value="adjective"/>
                                          <enumeration value="verb"/>
                                          <enumeration value="adverb"/>
                                          <enumeration value="conjunction"/>
                                          <enumeration value="preposition"/>
                                          <enumeration value="postposition"/>
                                          <enumeration value="article"/>
                                          <enumeration value="interjection"/>
                                       </restriction>
                                    </simpleType>
                                 </list>
                              </simpleType>
                              <simpleType>
                                 <list itemType="mpeg7:termReferenceType"/>
                              </simpleType>
                           </union>
                        </simpleType>
                     </attribute>
                     <attribute name="baseForm" type="string" use="optional"/>
                     <attribute      name="phonogrammicRepresentation"      type="string"
use="optional"/>
                  </extension>
               </simpleContent>
            </complexType>
         </element>
      </choice>
      <attribute name="identifier" type="ID" use="optional"/>
      <attribute name="equal" type="IDREF" use="optional"/>
      <attribute name="operator" use="optional">
         <simpleType>
            <union memberTypes="mpeg7:dependencyOperatorType mpeg7:termReferenceType"/>
         </simpleType>
      </attribute>
      <attribute name="particle" type="string" use="optional"/>
      <attribute name="synthesis" use="optional" default="dependency">
         <simpleType>
            <restriction base="NMTOKEN">
               <enumeration value="dependency"/>
               <enumeration value="coordination"/>
            </restriction>
         </simpleType>
      </attribute>
   </complexType>
```

Within the AXMEDIS project a MPEG-7 descriptor especially for text will be developed. This descriptor reflects specific information that are relevant for text resource and that can be extracted automatically. Relevant features, which have to be considered, are:

- size
- character encoding
- number of characters
- number of words
- number of lines
- number of periods
- number of paragraphs
- words
- lines
- periods

15.2 Audio Descriptors defined in MPEG-7 (EPFL):

Audio content may be classified according to some arbitrary taxonomy. For example, one may need to discriminate music from speech and general sounds (other than music or speech). Audio content categorised as music may be further defined in terms of musical genres while speech content may be classified in terms of voice type (male, female, child). Note that this framework may be used to classify audio content following other dimensions such as instrumentation (*guitars*, *strings*, *keyboards*, *brasses*…), mood (*aggressive*, *dark*, *dramatic*, *exotic*, *funky*, *futuristic*, *lonely*, *romantic*…) or recording type (*studio*, *live*).

MPEG-7's ClassificationScheme DS defines a set of language-independent terms that can be used for classifying some subject area. It also can organize the terms by establishing relationships amongst those terms. A ClassificationScheme DS is made up of a set of Items, each defining one term in the classification scheme. Each Item includes a unique identifier for a term (used to reference it via the term attribute), a set of human readable labels for the term, and a set of human readable definitions of what the term means.

Here is the syntax of the ClassificationScheme DS:

```
<!-- ################################################   -->
<!-- Definition of ClassificationScheme DS              -->
<!-- ################################################   -->

<complexType name="ClassificationSchemeType">
   <complexContent>
      <extension base="mpeg7:DSType">
         <sequence maxOccurs="1" minOccurs="1">
            <element name="Description" type="mpeg7:TextualType"
               minOccurs="0" maxOccurs="unbounded"/>
            <choice minOccurs="1" maxOccurs="unbounded">
               <element name="Item" type="mpeg7:ItemType"
                  minOccurs="1" maxOccurs="1"/>
               <element name="ItemImport"
                  type="mpeg7:ItemImportType"
                  minOccurs="1" maxOccurs="1"/>
               <element name="ClassificationSchemeRef"
                  type="mpeg7:ClassificationSchemeRefType"
                  minOccurs="1" maxOccurs="1"/>
               </choice>
         </sequence>
         <attribute name="scheme" type="mpeg7:classificationSchemeIdentifierType"
            use="required"/>
         <attribute name="mpeg7id" type="string" use="optional"/>
         <attribute name="version" type="string" use="optional"/>
      </extension>
   </complexContent>
</complexType>

<!-- ################################################ -->
<!-- Definition of Item datatype                      -->
<!-- ################################################ -->
```

```
<complexType name="ItemType">
   <sequence minOccurs="1" maxOccurs="1">
      <element name="Label" minOccurs="0" maxOccurs="unbounded">
         <complexType>
            <simpleContent>
               <extension base="mpeg7:TextualType">
                  <attribute name="preferred" type="boolean" use="optional"/>
               </extension>
            </simpleContent>
         </complexType>
      </element>
      <element name="Definition" type="mpeg7:TextualType"
         minOccurs="0" maxOccurs="unbounded"/>
      <choice minOccurs="1" maxOccurs="unbounded">
         <element name="Item" minOccurs="1" maxOccurs="1">
            <complexType>
               <complexContent>
                  <extension base="mpeg7:ItemType">
                     <attribute name="type"  type="mpeg7:termRelationKindType"
                        use="default" value="NT"/>
                  </extension>
               </complexContent>
            </complexType>
         </element>
         <element name="ItemImport" type="mpeg7:ItemImportType"
            minOccurs="1" maxOccurs="1"/>
         <element name="ClassificationSchemeRef"
            type="mpeg7:ClassificationSchemeRefType"
            minOccurs="1" maxOccurs="1"/>
      </choice>
   </sequence>
   <attribute name="term" type="mpeg7:controlledTermIdentifierType" use="required"/>
</complexType>

<!-- ################################################ -->
<!-- Definition of ItemImport datatype                -->
<!-- ################################################ -->

<complexType name="ItemImportType">
   <complexContent>
      <extension base="mpeg7:ItemType">
         <attribute name="importScheme" type="QName" use="required"/>
         <attribute name="importTerm" type="string" use="required"/>
      </extension>
   </complexContent>
</complexType>

<!-- ################################################   -->
<!-- Definition of ClassificationSchemeRef              -->
<!-- ################################################   -->

<complexType name="ClassificationSchemeRefType">
   <attribute name="schemeLocation" type="mpeg7:classificationSchemeLocatorType"
      use="optional"/>
   <attribute name="scheme" type="mpeg7:classificationSchemeIdentifierType"
      use="required"/>
</complexType>
```

Here is the semantics of the ClassificationScheme DS:

- Semantics of the ClassificationSchemeType:

Name	*Definition*
ClassificationSchemeType	Description scheme defining a set of terms and their relations.

Name	*Definition*
Description	Indicates a human readable explanation of the classification scheme.
Item	Describe one item in this classification scheme.
ItemImport	Describes one imported from another existing classification scheme. This allows new classification schemes to extend and build onto existing classification schemes.
ClassificationSchemeRef	References a non-external classification from which all terms are to be incorporated into this classification scheme.
Scheme	Identifies the classification with a fully qualified name of the classification scheme. The namespace URL associated with this identifier should reference the authoritative definition for the classification, if one exists.
mpeg7id	Indicates the MPEG-7 description tool(s) to which this classification scheme applies to using an XPath expression. The path is defined relative to an MPEG-7 *description*, **not** the DDL schema definition. For example: the value "//ClassificationScheme/@scheme" would indicate that the tool is appropriate for "scheme" attribute of the description tool represented by a "ClassificationScheme" element.
Version	Identifies the version of the classification scheme. The contents of this field are not specified by MPEG-7; however, the same version must always be identified with the same string.

- Semantics of the ItemType:

Name	*Definition*
ItemType	Datatype representing a single term definition in the ClassificationScheme DS.
Label	Indicates the human readable label for the item. It is possible to have multiple labels for the same item, possibly in different languages. The languages of a label is indicated by the xml:lang attribute.
preferred	Indicates whether or not this is the preferred label for this term. In the case where multiple labels exist, only one label per language shall be marked preferred.
Definition	Indicates the human readable explanation of the item. It is possible to have multiple definitions in different languages for the same item. The languages of a description is indicated by the xml:lang attribute.
Item	Describes a set of items related to the containing item.
Type	Indicates the type of relation existing between the contained item and the containing item. By default, it is "NT", indicating that the contained item is narrower in meaning than this item.
ClassificationSchemeRef	A reference to a ClassificationScheme that is inserted at the current level of the hierarchy. See below for detailed explanation of including classification schemes via referencing.
Term	The unique identifier for this term in the classification scheme. All items within a single classification must be unique.

- Semantics of the ItemImportType:

Name	*Definition*
ItemImportType	Datatype describing a term imported from an existing classification scheme.
Term	Identifies the term in the current classification scheme. It may be different from the imported term's original identifier, which is designated by importTerm.
importScheme	The identifier of the classification scheme from which the term is being imported.
importTerm	Identifies the term identifier in the classification scheme from which is being imported.

- Semantics of the ClassificationSchemeRefType:

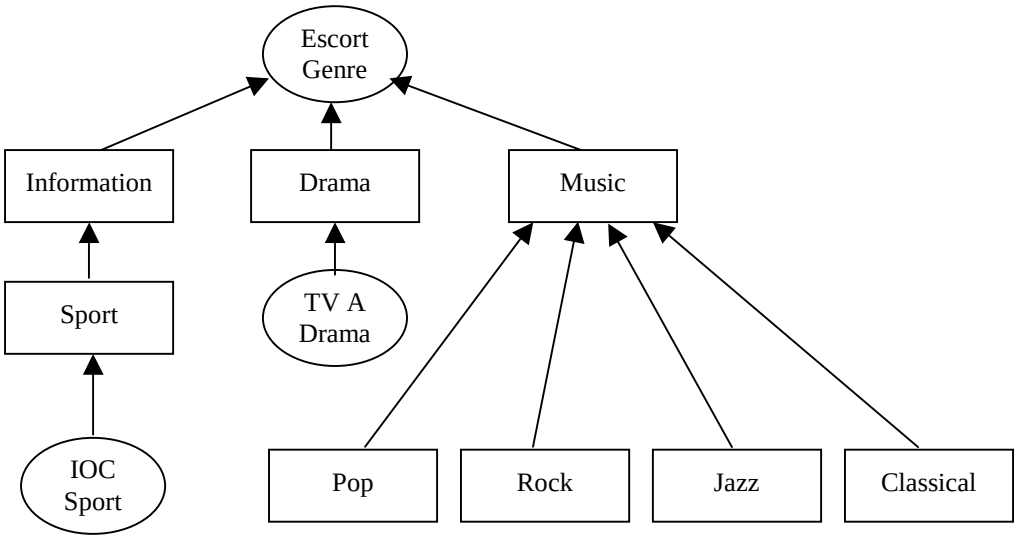
Name	*Definition*
ClassificationSchemeRefTyp e	A reference to the ClassificationScheme DS.
Scheme	Identifies the classification scheme being referenced.
schemeLocation	References the location of the definition of the referenced classification scheme.

One classification may be imported into another using ClassificationSchemeRef. This is useful when one wants to combine several independent classification schemes into a single larger classification scheme.

Let ReferencedCS refer to the classification scheme referenced by the definition of classification scheme DefineCS. An item that is not contained within another item definition is called a top-level item. Then the reference to a classification scheme within the definition of another classification scheme is interpreted as follows.

- The set of items defined in ReferencedCS is added to the set of items in DefineCS. It is an error if an item with the same value for term occurs in both ReferencedCS and DefineCS.
- The value of term, and the values the labels and definitions are incorporated unmodified into DefineCS from ReferencedCS.
- The set of term relations defined in ReferncedCS is added to the set of term relations in DefineCS.
- If the ClassificationSchemeRef is at the topmost level of DefineCS, then the top-level elements in ReferencedCS are added to the set of top-level item in DefineCS.
- If the ClassificationSchemeRef is not at the topmost level - i.e. occurs within an item definition - then all top-level elements of ReferencedDS are related to the containing term by a "narrower term" relation.

The following figure shows an example of a simple classification for genre identified as "Escore:Genre2.4". In the figure round boxes represent classification schemes and square boxes items.



The "Escort:Genre2.4" Genre Classification Scheme.

In this example, there are three items at the highest level: information, drama, and music. Under the information category there a more detailed term: sports. Rather than defining a complete classification for sports, this example shows how an existing classification can be "spliced" into the classification scheme

hierarchy. In this case the existing "IOC Sports" classification scheme is added under the Sports item. Similarly, the drama category includes the "TVE Drama" classification scheme. For music, the sub-items (narrower in meaning than their containing item) are "pop", "rock", "jazz", and "classical".

```
<ClassificationScheme
      scheme="Escort2_4:Content"
      mpg7id="CreationInformation/Classification/Genre">

      <Item term="1">
            <Label xml:lang="en">Information</Label>
            <Definition xml:lang="en">Generic news</Definition>

            <Item term="1.1">
                  <Label xml:lang="en">Sport</Label>
                  <Definition xml:lang="en">Sports news</Definition>

                  <ClassificationSchemeRef
                        xmlns:IOC="http://www.ioc.org"
                        scheme="IOC:Sports">
            </Item>
      </Item>

      <Item term="2">
            <Label xml:lang="en">Drama</Label>
            <Definition xml:lang="en">Dramatic Programs</Definition>

            <ClassificationSchemeRef
                  xmlns:TVE = "http://www.tvid.org"
                  scheme="TVE:Drama"/>
      </Item>

      <Item term="3">
            <Label xml:lang="en">Music</Label>
            <Definition xml:lang="en">Musical Programs</Definition>
            <Item term="3.1">
                  <Label xml:lang="en">Rock</Label>
            </Item>
            <Item term="3.2">
                  <Label xml:lang="en">Pop</Label>
            </Item>
            <Item term="3.3">
                  <Label xml:lang="en">Jazz</Label>
            </Item>
            <Item term="3.4">
                  <Label xml:lang="en">Classical</Label>
            </Item>
      </Item>
</ClassificationScheme>
```

Content navigation and access: structure of audio content and summarization

This section provides MPEG-7 description schemes helping in browsing and navigating into the audio content. It provides schemes to store audio segments representative of the structure of the content. For example, the typical structure of a pop song may look like: intro verse, chorus, second verse, chorus, bridge, third verse, chorus, coda, outtro. In the case of spoken content, one may wish to structure the audio content in terms of speakers or subject.

The HierarchicalSummary DS is constructed around the generic notion of temporal segments of AV data, described by HighlightSegments. Each HighlightSegment contains locators to the AV data, to provide access to the associated key-videoclip or key-audioclip, to key-frames and to key-sounds and may also contain textual annotation referring to key-themes. These audiovisual segments are grouped into summaries, or highlights, using the HighlightSummary description scheme. Such summaries may correspond to two different themes and could provide alternative views on the original AV content. The HighlightSummary description scheme is recursive in nature, enabling summaries to contain other summaries. This capability can be used to build a variety of hierarchical summaries, i.e. to describe content at different granularities. Additionally, multiple summaries may be grouped together using the HierarchicalSummary description scheme.

Here is the syntax of the description scheme involved in the summarization definition:

```
<!-- ################################################ -->
<!-- Definition of Summarization DS                   -->
<!-- ################################################ -->

<complexType name="SummarizationType">
   <complexContent>
      <extension base="mpeg7:DSType">
         <sequence>
            <element name="Summary" type="mpeg7:SummaryType"
               minOccurs="1" maxOccurs="unbounded"/>
         </sequence>
      </extension>
   </complexContent>
</complexType>

<!-- ################################################ -->
<!-- Definition of Summary DS                         -->
<!-- ################################################ -->

<complexType name="SummaryType" abstract="true">
   <complexContent>
      <extension base="mpeg7:DSType">
         <sequence>
            <element name="Name" type="mpeg7:TextualType"
               minOccurs="0" maxOccurs="1"/>
            <element name="SourceLocator" type="mpeg7:MediaLocatorType"
               minOccurs="0" maxOccurs="1"/>
            <element name="SourceInformation" type="mpeg7:ReferenceType"
               minOccurs="0" maxOccurs="1"/>
         </sequence>
      </extension>
   </complexContent>
</complexType>

<!-- ################################################ -->
<!-- Definition of HierarchicalSummary DS             -->
<!-- ################################################ -->

<complexType name="HierarchicalSummaryType">
   <complexContent>
      <extension base="mpeg7:SummaryType">
         <sequence>
            <element name="SummaryThemeList" type="mpeg7:SummaryThemeListType"
               minOccurs="0" maxOccurs="1"/>
            <element name="HighlightSummary" type="mpeg7:HighlightSummaryType"
               minOccurs="1" maxOccurs="unbounded"/>
         </sequence>
         <attribute name="components" use="required">
            <simpleType>
               <list>
                  <simpleType>
                     <restriction base="string">
```

```
                        <enumeration value="keyVideoClips"/>
                        <enumeration value="keyAudioClips"/>
                        <enumeration value="keyFrames"/>
                        <enumeration value="keySounds"/>
                        <enumeration value="keyThemes"/>
                     </restriction>
                  </simpleType>
               </list>
            </simpleType>
         </attribute>
         <attribute name="hierarchy" use="required">
            <simpleType>
               <restriction base="string">
                  <enumeration value="independent"/>
                  <enumeration value="dependent"/>
               </restriction>
            </simpleType>
         </attribute>
      </extension>
   </complexContent>
</complexType>

<!-- ################################################ -->
<!-- Definition of SummaryThemeList DS                -->
<!-- ################################################ -->

<complexType name="SummaryThemeListType">
   <complexContent>
      <extension base="mpeg7:DSType">
         <sequence>
            <element name="SummaryTheme" minOccurs="1" maxOccurs="unbounded">
               <complexType>
                  <simpleContent>
                     <extension base="mpeg7:TextualType">
                        <attribute name="id" type="ID"
                           use="required"/>
                        <attribute name="parentId" type="IDREF"
                           use="optional"/>
                     </extension>
                  </simpleContent>
               </complexType>
            </element>
         </sequence>
      </extension>
   </complexContent>
</complexType>

<!-- ################################################ -->
<!-- Definition of HighlightSummary DS                -->
<!-- ################################################ -->

<complexType name="HighlightSummaryType">
   <complexContent>
      <extension base="mpeg7:DSType">
         <sequence>
            <element name="Name" type="mpeg7:TextualType"
               minOccurs="0" maxOccurs="1"/>
            <element name="HighlightSegment" type="mpeg7:HighlightSegmentType"
               minOccurs="1" maxOccurs="unbounded"/>
            <element name="HighlightChild" type="mpeg7:HighlightSummaryType"
               minOccurs="0" maxOccurs="unbounded"/>
         </sequence>
         <attribute name="level" type="integer"
            use="optional"/>
         <attribute name="duration" type="mpeg7:mediaDurationType"
            use="optional"/>
         <attribute name="numKeyFrames" type="nonNegativeInteger"
            use="optional"/>
```

```
         <attribute name="fidelity" type="mpeg7:zeroToOneType"
            use="optional"/>
         <attribute name="themeIds" type="IDREFS"
            use="optional"/>
      </extension>
   </complexContent>
</complexType>

<!-- ################################################ -->
<!-- Definition of HighlightSegment DS                -->
<!-- ################################################ -->

<complexType name="HighlightSegmentType">
   <complexContent>
      <extension base="mpeg7:DSType">
         <sequence>
            <element name="Name" type="mpeg7:TextualType"
               minOccurs="0" maxOccurs="1"/>
            <element name="KeyVideoClip" type="mpeg7:VideoSegmentLocatorType"
               minOccurs="0" maxOccurs="1"/>
            <element name="KeyAudioClip" type="mpeg7:AudioSegmentLocatorType"
               minOccurs="0" maxOccurs="1"/>
            <element name="KeyFrame" type="mpeg7:ImageLocatorType"
               minOccurs="0" maxOccurs="unbounded"/>
            <element name="KeySound" type="mpeg7:AudioSegmentLocatorType"
               minOccurs="0" maxOccurs="unbounded"/>
         </sequence>
         <attribute name="themeIds" type="IDREFS" use="optional"/>
      </extension>
   </complexContent>
</complexType>
```

Here is the semantics of the different terms involved in the summarization definition:

Semantics of the SummarizationType:

Name	*Definition*
SummarizationType	Specifies a set of Summary elements.
Summary	An AV summary of AV content or a related group of summaries. See section **Errore. L'origine riferimento non è stata trovata.**.

Semantics of the SummaryType:

Name	*Definition*
SummaryType	An abstract DS from which the following description schemes are derived: 3. HierarchicalSummary DS 4. SequentialSummary DS
Name	Specifies the name of an instantiation of the Summary DS.
SourceLocator	Specifies location of the original (source) AV content that is summarized.
SourceInformation	References an element of a description of the original (source) AV content. Shall refer to a valid id attribute of a description element.

Semantics of the HierarchicalSummaryType:

Name	*Definition*
HierarchicalSummaryType	Specifies a group of summaries that contain hierarchically ordered audio-visual segments. A HierarchicalSummary element contains HighlightSummary elements, each of which specify a single, complete summary.

Name	*Definition*
SummaryThemeList	Specifies a list of textual themes associated with parts of the HierarchicalSummary.
HighlightSummary	Specifies a single AV summary, which can consist of a hierarchy of components. Each HighlightSummary represents an alternative view on the AV content. See section **Errore. L'origine riferimento non è stata trovata.**.
components	Specifies a list of the types of summary components included in a HierarchicalSummary. The types allowed are defined as follows. • *keyVideoClips* - The summaries shall contain key-videoclips, possibly ordered hierarchically. Such video clips form a video summary of a particular duration. A key-videoclip can be a video segment from the content, or from related media. • *keyAudioClips* - The summaries shall contain key-audioclips, possibly ordered hierarchically. Such audio clips form an audio summary of a particular duration. A key-audioclip can be an audio segment from the content, or from related media. • *keyFrames* - The summaries shall contain key-frames, possibly ordered hierarchically. A summary may contain a higher number of key-frames on each subsequent level of its hierarchy, to provide different levels of detail. A key-frame can be a specific frame from a video segment, or an image that is not in the video, possibly a synthetic image (pre-composed from multiple images). • *keySounds* - The summaries shall contain key-sounds, possibly ordered hierarchically. A summary may contain a higher number of key-sounds on each subsequent level of its hierarchy, to provide different levels of detail. A key-sound may correspond to key words in speech, sound effects, emotional sounds, exploding sounds, specific instrument sounds, and possibly synthetic sounds. • *keyThemes* - The summaries shall contain videoclips and/or audioclips, possibly ordered hierarchically, as well as textual descriptions of associated events or themes. Each summary is a collection of videoclips and/or audioclips referring to particular key-events or themes. Key-events or themes may be described textually by key-words.
hierarchy	Indicates the type of the hierarchy with respect to the parent-child relationships between elements at different levels of the hierarchy. This attribute may be used to eliminate unnecessary duplication of information in a hierarchy of elements. The types of the hierarchy are defined as follows. 5. *independent* - The information in the elements on a single level of a hierarchy completely specifies a particular summary, without reference to the information in the parent elements of these elements. Information in the parent elements shall not be re-used in the children elements. 6. *dependent* - The information in children elements in a hierarchy adds to, or refines, the information in the parent elements. Information in the parent elements shall be re-used in the children elements. Note that the value of this attribute may be ignored if none of the HighlightSummary elements of a HierarchicalSummary contain HighlightChild elements.

Semantics of the SummaryThemeListType:

Name	*Definition*
SummaryThemeListType	Defines a list of SummaryTheme elements.
SummaryTheme	Describes an event or theme in textual form, in terms of which a video can be summarized.
id	Identifies an instantiation of a SummaryTheme element.
parentId	Refers to another SummaryTheme element that corresponds to the parent- or super-theme in a conceptual hierarchy of themes (optional). Shall refer to the valid id attribute of a SummaryTheme element.

Semantics of the HighlightSummaryType:

Name	*Definition*
HighlightSummaryType	Specifies a single summary or part of a summary. Contains a set of audio-visual segments that form a summary. A HighlightSummary element may contain HighlightSummary elements as its children, in which case a tree-based hierarchy of summary elements may be formed. Each tree has a single root element that is part of a HierarchicalSummary element and all elements in a single tree correspond to the same summary (at different levels of detail).
Name	Name of a particular HighlightSummary element.
HighlightSegment	Describes an audio-visual segment by its key-videoclip and/or key-audioclip, its key-frames, key-sounds and key-themes. See section **Errore. L'origine riferimento non è stata trovata.**.
HighlightChild	Describes a child HighlightSummary element that describes the current summary in more detail. Child HighlightSummary elements are used to form a tree-based hierarchy of summary components. A summary at a particular level of detail is to be constructed by combining information from all HighlightChild nodes at or up to the same level in a single tree. If the hierarchyType of the hierarchical summary tree is "independent", the summary components (key-videoclips, key-audioclips, key-frames and key-sounds) in all children HighlightSummary elements at the same level of a single tree shall be combined to define a single AV summary. If the hierarchyType is "dependent", the summary components in all children HighlightSummary elements up to a particular level shall be added to the components of their parent tree nodes (recursively up the tree) to define a single, complete AV summary. In the latter case, all elements up to a particular level in the tree contribute to an AV summary at a particular level of detail.
level	Indicates the level of a HighlightSummary element in a hierarchy (optional). The root HighlightSummary element in a hierarchy has level 0, its children HighlightSummary elements have level 1, etc.
duration	Indicates the temporal duration of the HighlightSegments contained in a HighlightSummary element (optional). Indicates the total duration of key-videoclips in a video summary; indicates the total duration of key-audioclips in an audio summary.
numKeyFrames	Indicates the total number of key-frames contained in the set of HighlightSegment elements in a HighlightSummary element (optional).
fidelity	Indicates how well the information in the HighlightSummary element is represented by the information in its parent HighlightSummary element, on a numerical scale between 0.0 and 1.0 (optional). Values closer to 1.0 correspond to better representations of this element by the associated parent element, while values closer to 0.0 correspond to worse representations.
themeIds	A list of references to SummaryTheme identifiers indicating key-themes (key-events) common to all children HighlightSummary and HighlightSegment elements (optional). Shall refer to valid id attributes of SummaryTheme elements.

Semantics of the HighlightSegmentType:

Name	*Definition*
HighlightSegmentType	Specifies an audio-visual segment. May contain a video segment (key-videoclip), an audio segment (key-audioclip), images (key-frame) or sounds (key-sound).
Name	Identifies the segment by name.
KeyVideoClip	Specifies the location of a key-videoclip. A key-videoclip is an (audio-)visual segment of AV content, which can be used for navigation, browsing and summarization. See section **Errore. L'origine riferimento non è stata trovata.** for the definition of VideoSegmentLocator.
KeyAudioClip	Specifies the location of a key-audioclip. A key-audioclip is an audio segment of AV content, which can be used for navigation, browsing and summarization. See section **Errore. L'origine riferimento non è stata trovata.** for the definition of AudioSegmentLocator.
KeyFrame	Specifies the location of a key-frame. A key-frame is a single video frame in AV content, which can be used for navigation, browsing and summarization. See section **Errore. L'origine riferimento non è stata trovata.** for the definition of ImageLocator.
KeySound	Specifies the location of a key-sound. A key-sound is a single sound in AV content, which can be used for navigation, browsing and summarization. See section **Errore. L'origine riferimento non è stata trovata.** for the definition of AudioSegmentLocator.
themeIds	A list of references to SummaryTheme identifiers indicating key-themes (key-events) common to all children HighlightSummary and HighlightSegment elements (optional). Shall refer to valid id attributes of SummaryTheme elements.

15.3 Video Descriptors defined in MPEG-7 (FHGIGD):

The implementation of metadata extractors from video files within AXMEDIS is limited to integration of one or more available content description algorithms for video. On the one hand the selection depends on the requirements within AXMEDIS, the currently used descriptors, and the availability/accessibility existing libraries. On the other hand the availability of the algorithms determines the meta data extraction algorithms for video. As described in the previous section, the reference software for MPEG-7 video descriptors will be integrated within the AXMEDIS framework. Therefore, the XML descriptors as initially defined in the MPEG-7 standards are applied. The integration of further content description algorithms might require the development of new data structures that are compliant with the MPEG-7 data structures.

Below some examples for MPEG-7 visual descriptors are given.

Basic output data structures

In MPEG-7 visual descriptors are defined, which are primarily considered for the meta data extracted from videos. As videos consists of a (time) series of images the basic data type is “VisualTimeSeriesType”.

```
<complexType name=”VisualTimeSeriesType” abstract=”true”>
  <sequence>
    <element name=”TimeIncr” type=”mpeg7:mediaDurationType”/>
  </sequence>
  <attribute name=”offset” type=”mpeg7:mediaDurationType”
       use=”defaultoptional” valuedefault=”PT0S”/>
</complexType>
```

In MPEG-7, regular and irregular time series are distinguished depending on the (non-) constant interval size of the collected descriptors.

Regular Visual Time Series

```
<complexType name=”RegularVisualTimeSeriesType” final=”#all”>
  <complexContent>
```

```
        <extension base="mpeg7:VisualTimeSeriesType">
          <sequence>
            <element name="Descriptors" type="mpeg7:VisualDType"
                 minOccurs="1" maxOccurs="unbounded"/>
          </sequence>
        </extension>
      </complexContent>
    </complexType>
```

Semantics of the RegularVisualTimeSeries:

DescriptorID	This elementfield, which is only present in the binary representation, specifies a descriptor identifier. The descriptor identifier specifies specifies the descriptor type accommodated in the time series.
NumOfDescriptorsNum	This elementfield, which is only present in the binary representation, specifies the number of descriptor instances accommodated in the time series.
IsRandomAccess	This elementfield, which is only present in the binary representation, specifies the access mode, which is either: • random access if the flag is set to 1; in this case DescriptorLength and BitStuffing elements are present in the binary representation • no random access if the flag is set to 0; in this case no bit stuffing is allowed and descriptor instances are not padded, which means they may have different lengths
DescriptorLength	This field, which is only present in the binary representation, specifies the length of each descriptor instance in bytes. The value of this element is the size of the largest descriptor instance, aligned to a byte boundary by bit stuffing using 0-7 '1' bits.
TimeIncr	This element specifies the default time interval. The time interval is defined as an interval between descriptor locations. An interval that follows a descriptor is associated with the descriptor. The type of this element "mediaDurationType" is specified in ISO/IEC 15938-5.
IsOffset	This field, which is only present in the binary representation, signals the presence of the offset attribute. If it is equal to 1 (true) offset is present, if 0 (false) offset is not specified (i.e. default value should be used).
offset	This attribute specifies the offset, i.e., the interval between the starting time point of a given time span and the location of the first descriptor. The default value is zero (represented as "PT0S" in DDL). This attribute is illustrated as "Offset" in **Errore. L'origine riferimento non è stata trovata.**.
BitStuffing	This field, which is only present in the binary representation, specifies stuffing bits (Aa sequence of '1's) stuffing bits to align on the byte boundary. to align the descriptor to a byte boundary.
Descriptors	This element contains the instantiation of specifies the visual descriptor accommodated in this time series. Only one type of child descriptor is allowed to be instantiated. Its binary syntax and semantics follow those of the assigned descriptor. In random access mode, if the size of a particular descriptor instance is smaller than DescriptorLength, it is padded with the required number of '1' bits.

Irregular Visual Time Series

```
<complexType name="IrregularVisualTimeSeriesType" final="#all">
   <complexContent>
      <extension base="mpeg7:VisualTimeSeriesType">
         <sequence minOccurs="1" maxOccurs="unbounded">
            <element name="Descriptors" type="mpeg7:VisualDType"/>
            <element name="Interval" type="mpeg7:unsigned32"/>
         </sequence>
      </extension>
   </complexContent>
</complexType>
```

Semantics of IrregularVisualTimeSeries:

DescriptorID	This elementfield, which is only present in the binary representation, specifies a descriptor identifier. The descriptor identifier specifies the descriptor type accommodated in the time series.
NumOfDescriptorsNum	This elementfield, which is only present in the binary representation, specifies the number of descriptor instances accommodated in the time series.
IsRandomAccess	This elementfield, which is only present in the binary representation, specifies the access mode, which is either: • random access if the flag is set to 1; in this case DescriptorLength and BitStuffing elements are present in the binary representation • no random access if the flag is set to 0; in this case no bit stuffing is allowed and descriptor instances are not padded, which means they may have different lengths
DescriptorLength	This elementfield, which is only present in the binary representation, specifies the length of each descriptor instance in bytes. The value of this element is the size of the largest descriptor instance, aligned to a byte boundary by bit stuffing using 0-7 '1' bits.
IsShortInterval	This elementfield, which is only present in the binary representation, indicates the size of the ShortInterval/LongInterval field. 1 (true) for 8-bit unsigned integer("unsigned8") while 0 (false) for 32-bit unsigned integer("unsigned32"). If IsShortInterval is set to 1, then 8-bit unsigned integer ("unsigned8") is used. If IsShortInterval is set to 0, then a 32-bit unsigned integer ("unsigned32") is used.
TimeIncr	This element specifies the base unit of the time interval. The time interval between descriptor locations is specified as a multiple of this base unit. The type of this element, MediaDurationType, is specified in ISO/IEC 15938-5.
IsOffset	This elementfield, which is only present in the binary representation, signals the presence of the offset attribute. If it is equal to 1 (true) offset is present, if 0 (false) offset is not specified (i.e. default value should be used).If IsOffset is set to 1 then the offset attribute is present. If IsOffset is set to 0 then the offset attribute is not specified (i.e. the default value should be used).
offset	This attribute specifies the offset, i.e., the interval between the starting time point of a given time span and the location of the first descriptor. The default value is zero (represented as "PT0S" in DDL). This element is illustrated as "Offset" in **Errore. L'origine riferimento non è stata trovata.**.
BitStuffing	This field, which is only present in the binary representation, specifies stuffing bits (a sequence of '1's) to align the descriptor to a byte boundary.

Descriptors	This element contains the instantiation of specifies the visual descriptor accommodated in this time series. Only one type of child descriptor is allowed to be instantiated. Its binary syntax and semantics follow those of the assigned descriptor. In random access, if the size of a particular descriptor instance is smaller than DescriptorLength, it is padded with the required number of '1' bits.
Interval/ShortInterval/LongInterval	This element specifies the time interval between the current and the preceding descriptor. The value of the element is specified in units defined by TimeIncr.

Within MPEG-7 feature descriptors are already defined including

- color,
- texture,
- shape, and
- motion.

MPEG-7 Colour descriptors

For the description of colour MPEG-7 includes several descriptors like:

5. **ColorSpace** is a supporting tool to express in which colour space the colour descriptors are expressed.
6. **ColorQuantization** is also a supporting and provides a mapping from the floating point values to an integer representation.
7. **DominantColor** specifies a set of dominant colours and targets content based retrieval for colors.
8. **ScalableColor** specifies a colour distribution .
9. **ColorLayout** specifies a global spatial colour distribution .
10. **ColorStructure** specifies a local spatial colour distribution .

MPEG-7 Texture descriptors

Textures so far are described in MPEG-7 by:

- **HomogeneousTexture** describes region texture by a frequency specific energy and energy deviation.
- **TextureBrowsing** specifies perceptual characterization of a texture (like regularity, coarseness, and directionality).
- **EdgeHistogram** specifies the spatial distribution of edges in local regions (sub-images).

MPEG-7 Shape descriptors

The already defined shape descriptors in MPEG are:

- **RegionShape** specifies a region-based shape of an object.
- **ContourShape** specifies a closed contour of a 2D object or region.
- **Shape3D** specifies the intrinsic shape description for 3D mesh models.

Motion

Different kinds of motion descriptors are already defined in MPEG-7:

- **CameraMotion** specifies 3D camera motion parameters.
- **MotionTrajectory** specifies the motion trajectory of a moving object.

- **ParametricMotion** specifies the motion of objects in video sequences.
- **MotionActivity** captures the notion of "intensity of motion" in a video segment (intensity of activity, direction of activity, spatial distribution of activity, spatial localization of activity, and temporal distribution of activity).

15.4 Content Descriptors for General Digital Resources (FHGIGD):

Only limited metadata information can be extracted from general digital resources as these files are only considered as binary files. Typical information, which might be relevant but cannot extracted from the digital resource directly include:

- Resource name
- Creation date
- Last modification date

In contrast to the previous information the only information that can be extracted automatically is

- resource size,
- cryptographic hash, and
- bit value distribution related information.

However, only the resource size and the corresponding cryptographic hash function are reasonable.

In MPEG-7 this kind of meta-information (for verification of the digital resource) is not consider so far. Ideally, each description should contain a cryptographic hash function to ensure the link between the object and meta-data. This is not yet foreseen within MPEG-7:

```
<!-- ################################################ -->
<!--  Definition of Basic Description Type                    -->
<!-- ################################################ -->

<complexType name="BasicDescriptionType" abstract="true">
   <sequence>
      <element name="Relationships" type="mpeg7:GraphType"
         minOccurs="0" maxOccurs="unbounded"/>
   </sequence>
</complexType>
```

Within AXMEDIS it has to be identified, how cryptographic hash values can be integrated best into MPEG-7 and the relationship with MPEG-21.

16 Fingerprint Estimation for Text files (DIPITA)

Module/Tool Profile Fingerprint Estimation for Text files		
Responsible Name	Zini	
Responsible Partner	DIPITA	
Status (proposed/approved)	Proposed	
Implemented/not implemented	Not implemented	
Status of the implementation	10%	
Executable or Library/module (Support)	Library	
Single Thread or Multithread	Single thread	
Language of Development	C++	
Platforms supported	MS WINDOWS	
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/newrepos/Framework/{source,include,project}/fingerprint/document/ /	
Reference to the AXFW location of the demonstrator executable tool for internal download		
Reference to the AXFW location of the demonstrator executable tool for public download		
Address for accessing to WebServices if any, add accession information (user aNd Passwd) if any		
Test cases (present/absent)	absent	
Test cases location		
Usage of the AXMEDIS configuration manager (yes/no)	no	
Usage of the AXMEDIS Error Manager (yes/no)	no	
Major Problems not solved		
Major pending requirements		
Interfaces API with other tools, named as	Name of the communicating tools References to other major components needed	Communication model and format (protected or not, etc.)
Formats Used	Shared with	format name or reference to a section
Plain Text		
Protocol Used	Shared with	Protocol name or reference to a

		section
Used Database name		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not

16.1.1 General description of the module

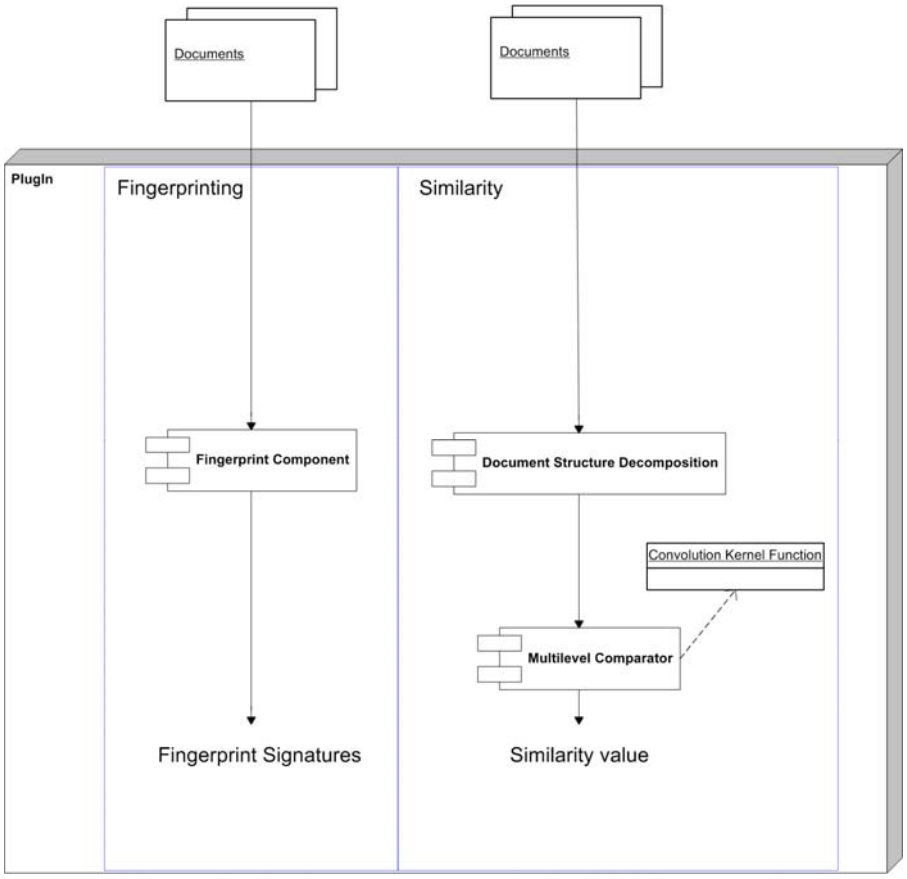
The text fingerprint plug-in output will be a string in which the value is stored.
Moreover the document comparison functionality also implemented in the plug-in will give as a result a normalized floating point value representing the degree of similarity between two given input documents.

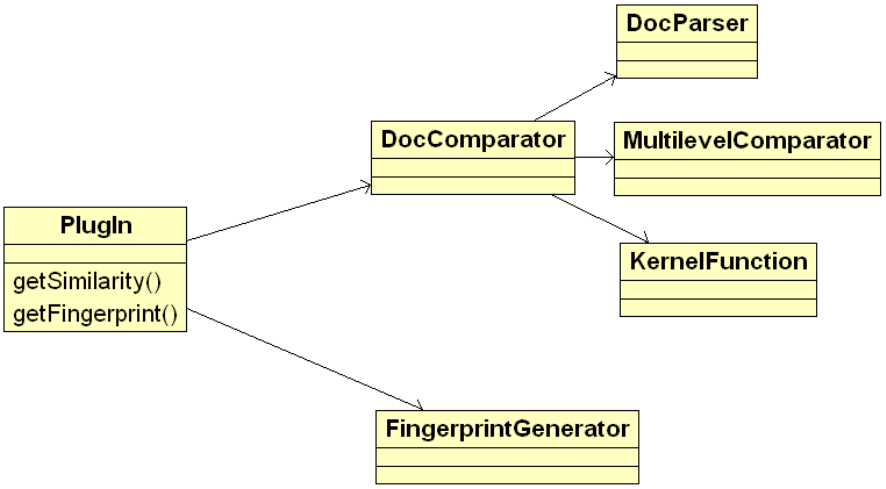
The text fingerprints plug-in aim is twofold, it provides a way of calculating a fingerprint value of the documents provided as input, moreover it provides functionalities for similarity estimation between two documents without making prior assumptions on the language.
The fingerprint algorithm hashes the ASCII representation of the input file and gives as result a string in which the fingerprint value is stored (full or selective hash values based on the analyzed document structure).

For similarity comparison a plug-in specific function is provided. This functionality could be exploited by several use cases including: identification of content from a sub part or when the different formats comparison is not straightforward, plagiarism detection and so on. The algorithm allows for robust multilevel comparison of documents taking into account document structure and leveraging the plagiarist behaviour, which is modeled as a combination of 3 basic actions: insertion, deletion, substitution. We recognize that this behavior may occur at various level of the document structure: the plagiarist may insert, delete or substitute a word, period or a paragraph. The procedure consists in two main steps: document structure extraction and plagiarism function calculation. We propose a recursive plagiarism evaluation function to be evaluated at each level of the document structure which is based on the Levenshtein edit distance.

For what concerns fingerprint it has to be said that in MPEG-7 this kind of meta-information (for verification of the text documents) is not consider so far. Within AXMEDIS it has to be identified, how cryptographic hash values can be integrated best into MPEG-7 and the relationship with MPEG-21.

16.1.2 Module Design in terms of Classes





16.1.3 Formal description of algorithm

Fingerprinting
The fingerprint algorithm hashes the ASCII representation of the input file and gives as result a string in which the fingerprint value is stored

Similarity and plagiarism extimation

3. Eliminate noise chars
4. Document graph construction (sentences, paragraphes, etc.)

5. Multilevel Similarity Function evaluation (The similarity function takes into account the plagiarist behavior and the graph structure)
6. Returns a double precision floating point normalized between 0 and 1

17 Fingerprint Estimation for Audio files (FHGIGD)

17.1 AudioID (property of m2any)

AudioID, an audio identification technology provided by m2any (http://www.m2any.de), is part of the MPEG-7 standard and is capable of recognizing a piece of music in a split second. The technology was developed at the renowned Fraunhofer Institute for Digital Media Technology (IDMT), well-known as the co-inventor of the MP3 format.

AudioID technology is essential for future consumption, monitoring and distribution of digital music by any method and in any format

The fields of applications for the AudioID technology are vast. We have customers using AudioID in the applications we list below, but nothing prevents you from helping us add to the list. m2any and the Fraunhofer Institute are also continuously developing additional applications, including embedded software, that use the basics of AudioID to continue enlarging the realm of possibilities.

AudioID is an external plug-in. It will be integrated by FHGIGD.

17.1.1 General Description of the Module

The basic concept behind a fingerprinting system is to identify a piece of audio content by extracting a compact and unique signature from it (so-called content-based identification). In a training phase, such signatures are created from a set of known audio material, and finally stored in a database.

Unknown content can then be identified by comparing its signature with those contained in the database.

Performance of the AudioID System:

In order to assess the system's recognition performance, the registered audio items are subjected to a wide range of signal manipulations which influence the audio signal's quality (e.g. equalization, acoustic transmission or MP3 encoding/decoding).

Similar to human recognition behaviour, which is surprisingly tolerant even to bad sounding signal alterations, the system is designed to be robust against acoustic interference.

Depending on the type of signal distortion applied, the achieved recognition rates are typically better than 99% with a recognition speed (on standard PC hardware) several orders of magnitude faster than the audio playback time.

AudioID and MPEG-7 Audio:

The AudioID system relies on a description core which has been standardized within the new MPEG-7 Audio Standard. It has the following benefits:

- Identification relies on a published, open feature format rather than proprietary solutions.
- MPEG-7 based signatures are likely to be produced as part of the standard metadata package which will accompany future advanced media formats.
- Due to the exact and standardized specification of the descriptor, inter operability is guaranteed on a worldwide basis, i.e. every search engine relying on the MPEG-7 specification will be able to use compliant descriptions, wherever they may have been produced.

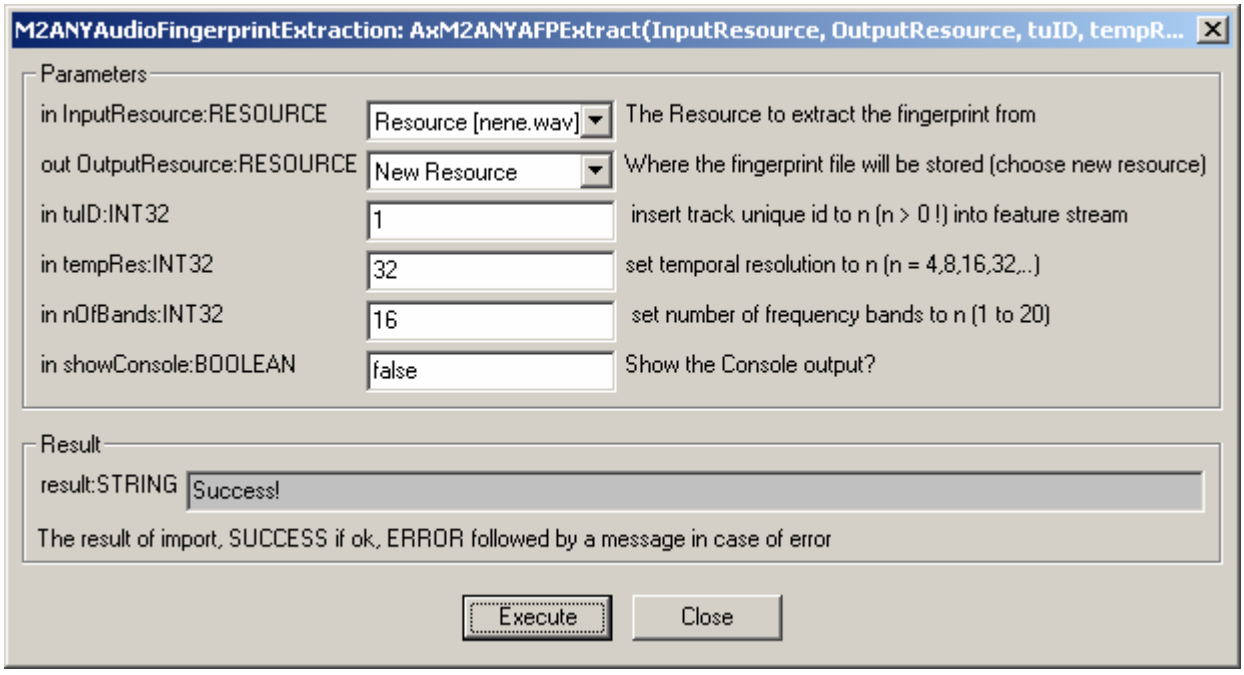
As a unique feature, AudioID MPEG-7 signatures are scalable, i.e. they allow a flexible trade-off between signature compactness and recognition robustness.

17.1.2 Module Design in terms of Classes

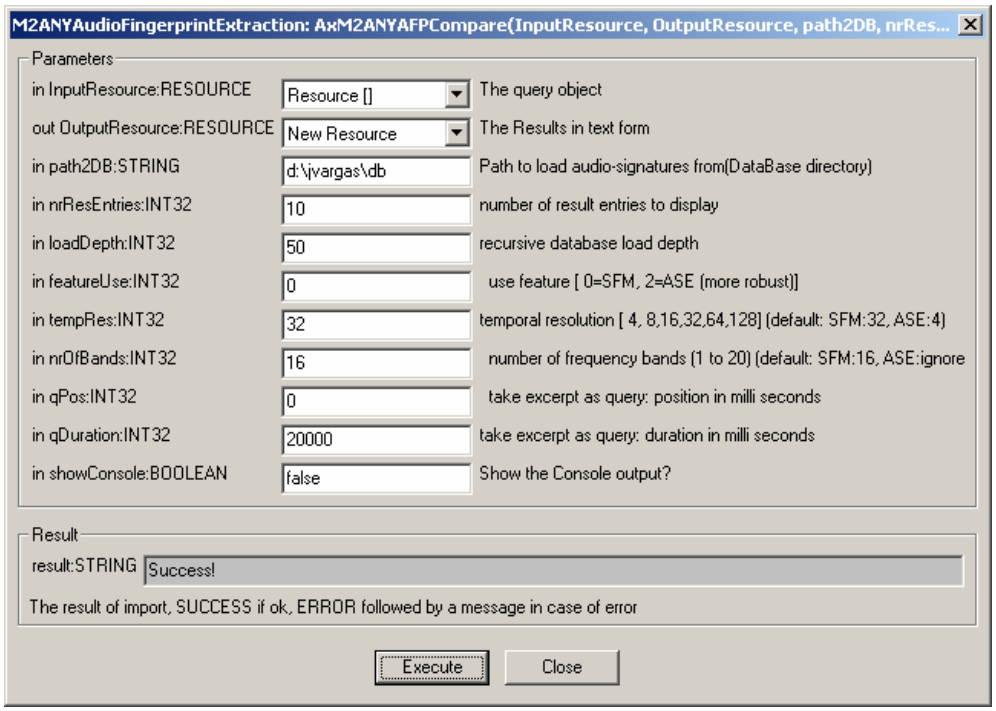
The main integration work consists of the implementation of a corresponding fingerprinting class, which calls the functionality developed within AXMEDIS.

17.1.3 User interface description

Usage of the developed plug-in depens on the AXMEDIs program that utilizes the available functionality. As shown in the next figure – which is an example when the plug-in is used in the AXEditor – several parameters can be set.



If the extracted fingerprint should be compared agains a database, the following parameters can be set:



17.1.4 Technical and Installation information

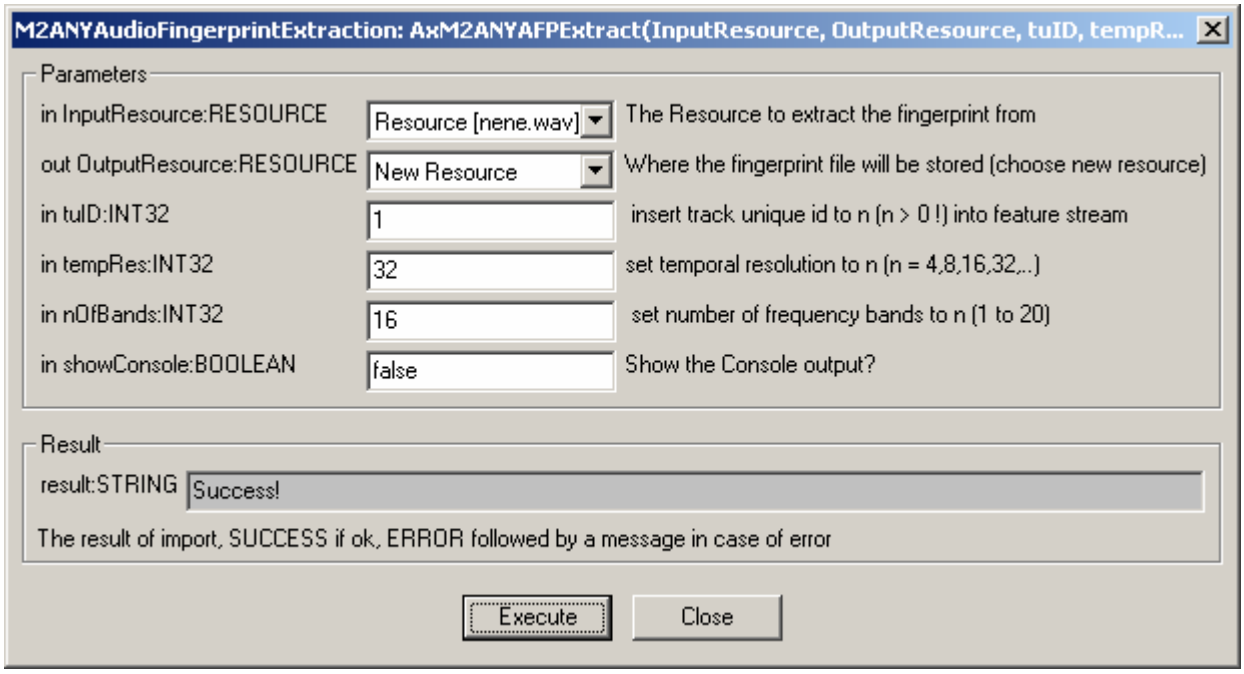
References to other major components needed	cfymain.exe, xtrmain.exe, cfy.dll, xtr.dll and asign.dll
Problems not solved	
Configuration and execution context	Plug-ins are installed by copying the library and the library description in the corresponding plug-in directory. You need your own copy of the needed files! Specially the executables (cfymain.exe, xtrmain.exe, cfy.dll, xtr.dll and asign.dll) installed in the corresponding plugin directory! You also need to indicate the path to an existing Fingerprint database.

17.1.5 Draft User Manual and Examples of Usage

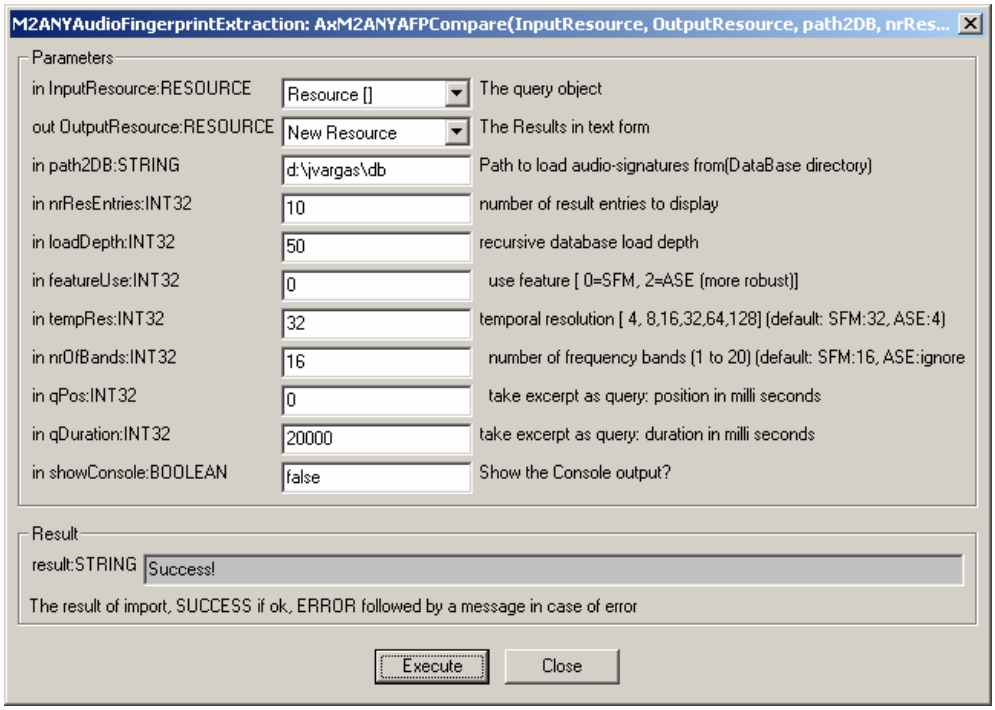
The plug-in can be applied to WAV files, provided this was declared in the mime type attribute of the resource. The output is a binary file containing the fingerprint in binary format. This example shows the usage of the developed plug-in with the AXEditor.

Before using the M2ANY audio fingerprinting an AXMEDIS object containing a wave resource has to be opened. Alternatively, a new AXMEDIS object can be created and a wave resource has to be added.

For the wave resource the "Content Processing Plug-in…" command has to be selected and the M2ANYAudioFingerprintExtraction plug-in has to be chosen.



To compare the resulted fingerprint against a fingerpint database, one has to select the "Content Processing Plug-in…" command on the "fingerprint/m2anyAFP"-mimetype-resource.



The resulting resource will be a text file. More detailed information is available in the plug-in description.

17.2 FIPSAudio

Module/Tool Profile	
FIPSAudio	
Responsible Name	Martin Schmucker (FHGIGD)
Responsible Partner	FHGIGD
Status (proposed/approved)	Proposed
Implemented/not implemented	Implemented
Status of the implementation	100%
Executable or Library/module (Support)	Library (PlugIn)
Single Thread or Multithread	Multithreaded
Language of Development	C++
Platforms supported	Microsoft Windows 32
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/newrepos/Framework/include/fingerprint/audio https://cvs.axmedis.org/newrepos/Framework/source/fingerprint/audio https://cvs.axmedis.org/newrepos/Framework/project/fingerprint/audio
Reference to the AXFW location of the demonstrator executable tool for internal download	https://cvs.axmedis.org/newrepos/Framework/bin/fingerprint/audio
Reference to the AXFW location of the demonstrator executable tool for public download	www.axmedis.org
Address for accessing to WebServices if any, add accession information (user and	

Passwd) if any		
Test cases (present/absent)	Absent	
Test cases location	-	
Usage of the AXMEDIS configuration manager (yes/no)		
Usage of the AXMEDIS Error Manager (yes/no)		
Major Problems not solved	-- --	
Major pending requirements	-- --	
Interfaces API with other tools, named as	Name of the communicating tools References to other major components needed	Communication model and format (protected or not, etc.)
Formats Used	Shared with	format name or reference to a section
Protocol Used	Shared with	Protocol name or reference to a section
Used Database name		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
FFMPEG		
FFTW	V3.0.1	GPL and Non-free license (see http://web.mit.edu/tlo/www/)
ImageMagick		

17.2.1 General Description of the Module

The Audio fingerprint plug-in is a tool that extracts an audio fingerprint of a given audio stream within a multimedia file. The audio stream can be embedded either in a normal audio file (mpg, wav, wma, etc…) or within a video file (mpeg, wmv, avi, etc…).

The Fingerprint extractors for audio files automatically calculate a digest describing its main characteristics in way suitable for automatic verification of AXMEDIS objects. Thus the descriptor is a low level description according to the previous definition.

Audio files are streams. These input streams is segmented. That means that for each segment a “sub-fingerprint” is calculated. Depending on the length of the input sequence, typically a request to the database does not result in a single fingerprint but in an array of fingerprints.

The characteristic processing steps are:

Feature extraction and processing: The input signal is pre-processed, which depends on the data type. For audio typical pre-processing operations are down-sampling, format conversion, and band-pass filtering. In the case of audio or video the input data are segmented and so-called "sub-fingerprints" are calculated. Features are generally extracted from a transformation domain. This transformation

domain redundancy is decreased (similar to compression). Within this transformation domain relevant features are extracted. In a post-processing specific relative measures can be derived.

Fingerprint modelling: The multi-dimensional input vector sequence is mapped to a single vector to produce compact fingerprints. This can also include a binarization.

To support numerous input formats, the audio fingerprinting plug-in is based on the FFMPEG library. Thus, the plug-in supports the same formats as supported by FFMPEG (see Appendix). The processing of the features is done in the Fourier domain (by using the FFTW library).

The integrated plug-in, which is based on the FFMPEG functionality licensed under LGPG, doesn't need to be configured. It was developed to be used in the AXMEDIS applications that allow the usage of AXMEDIS plug-ins. These include the AXEditor and the AXRuleEditor.

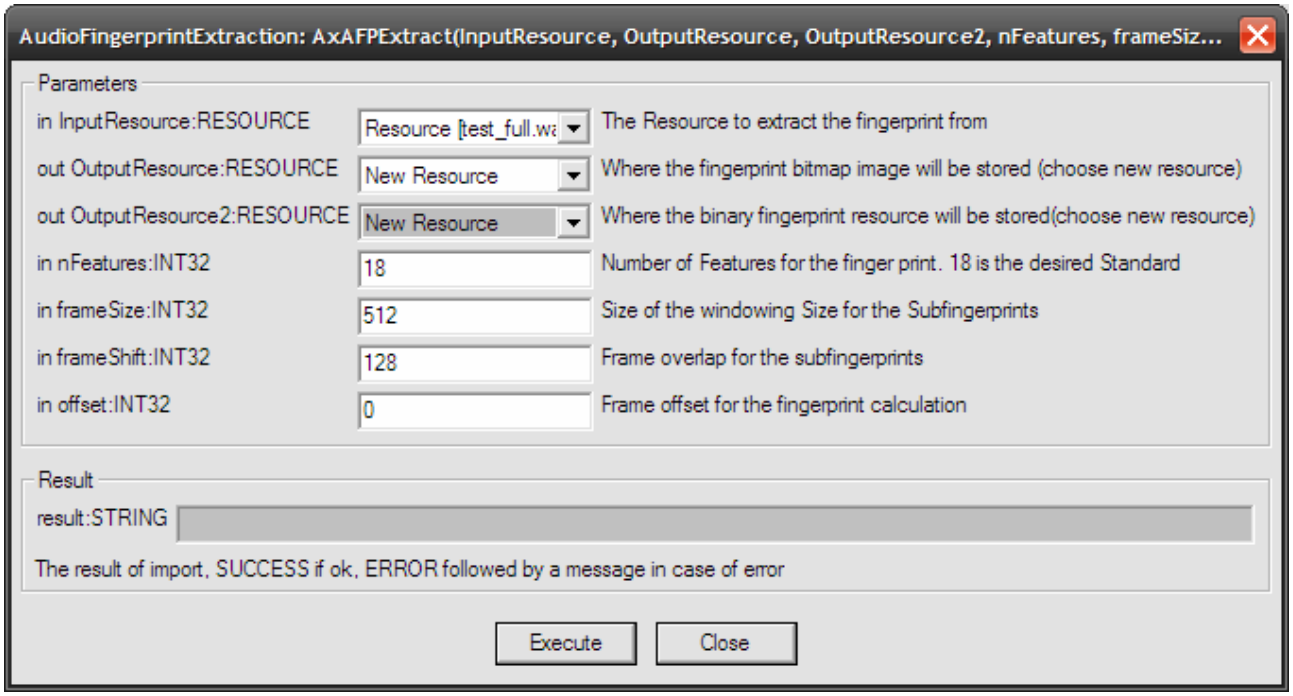
The implemented version was tested on MS Windows platform. Due to the platform independence of FFMPEG a conversion to other platforms is possible without spending too much effort on the core video adaptation functionality.

17.2.2 Module Design in terms of Classes

The main integration work consists of the implementation class `AudioFingerprinting`, which calls the functionality developed within AXMEDIS. This library is a wrapper class that calls the corresponding functionalities of the FFMPEG library.

17.2.3 User interface description

Usage of the developed plug-in depends on the AXMEDIS program that utilizes the available functionality. As shown in one of the next figures – which is an example when the plug-in is used in the AXEditor – several parameters can be set.



17.2.4 Technical and Installation information

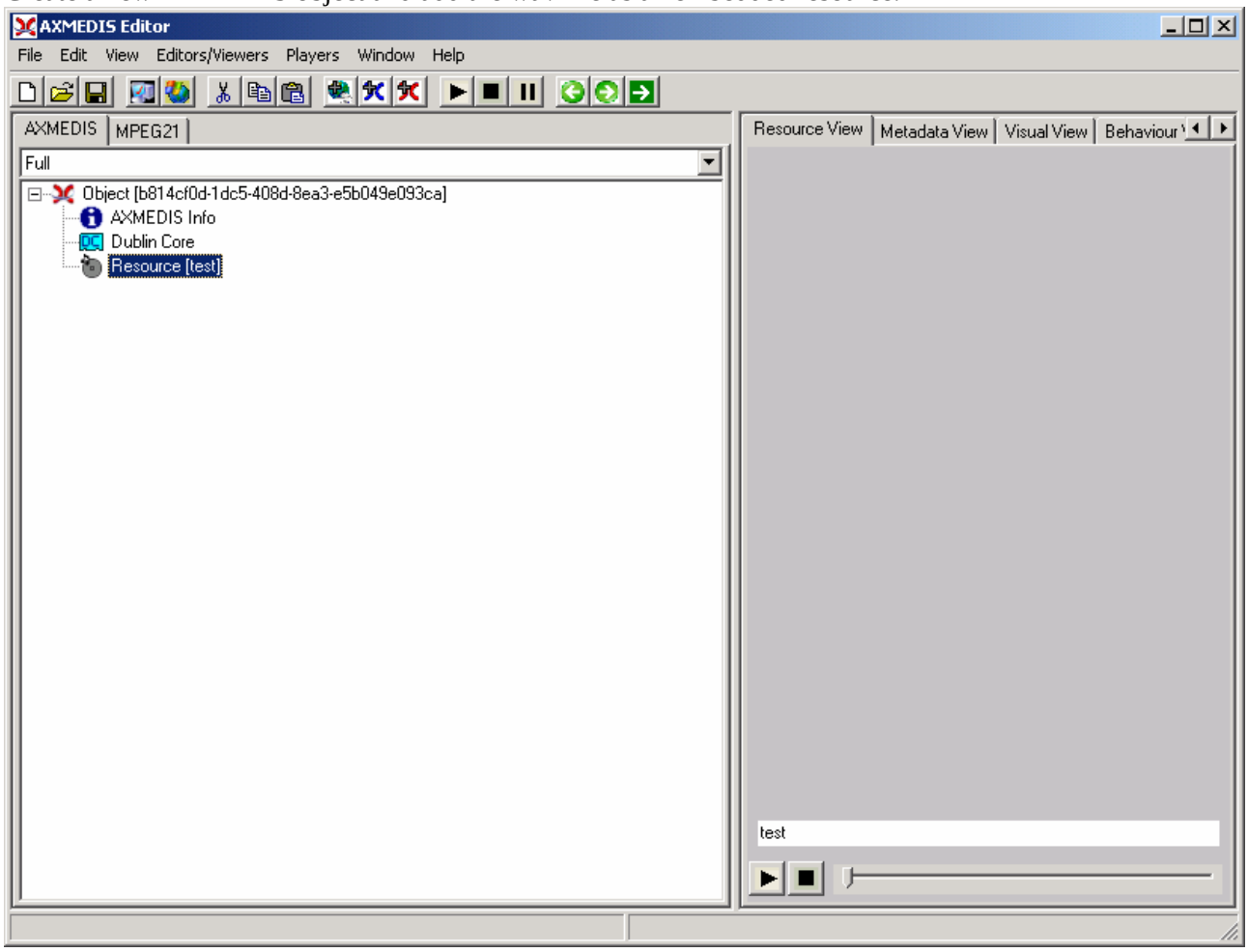
References to other major components needed	Dynamic Link Libraries (DLLs) of the FFMPEG library and ImageMagick for the visual output (avcodec-51_AXFP.dll, avformat-

	50_AXFP.dll and avutil-49_AXFP.dll).
Problems not solved	-
Configuration and execution context	Plug-ins are installed by copying the library and the library description in the corresponding plug-in directory.

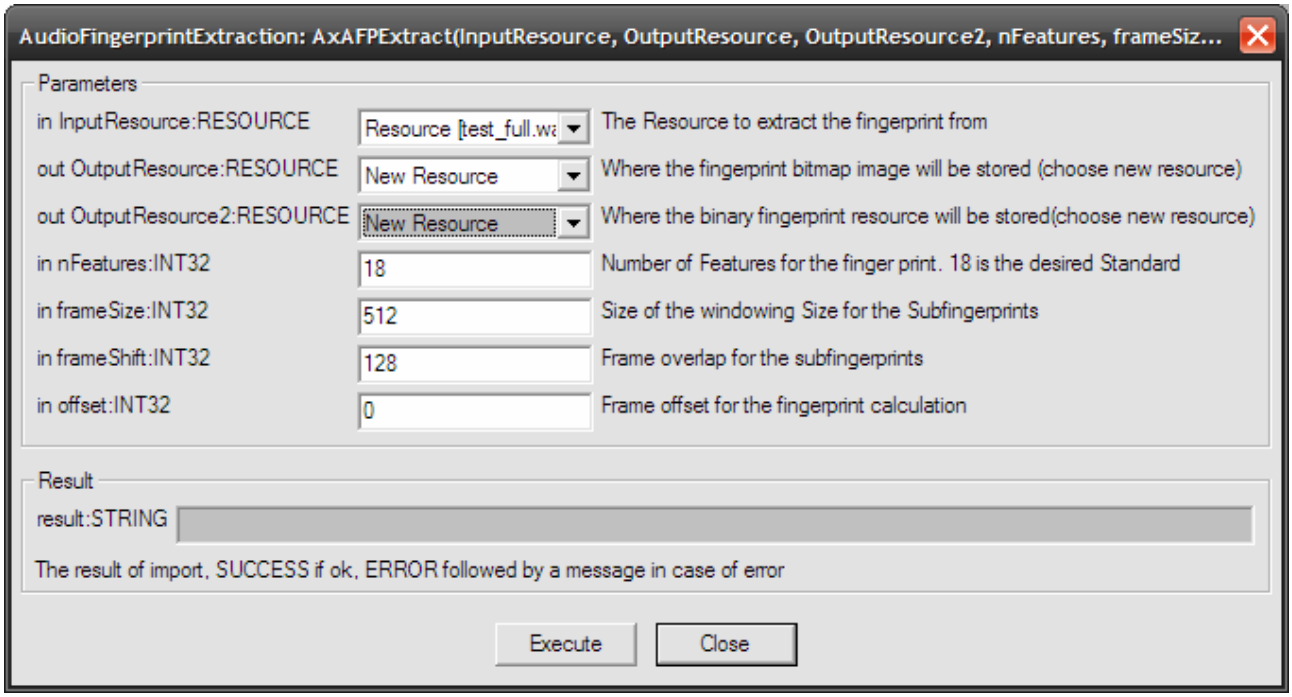
17.2.5 Draft User Manual and Examples of usage

The plug-in can be applied to any audio or video resources, provided this was declared in the mime type attribute of the resource. The output is of mimetype “image/x-ms-bmp”.

Create a new AXMEDIS object and add the wav file as an embedded resource.

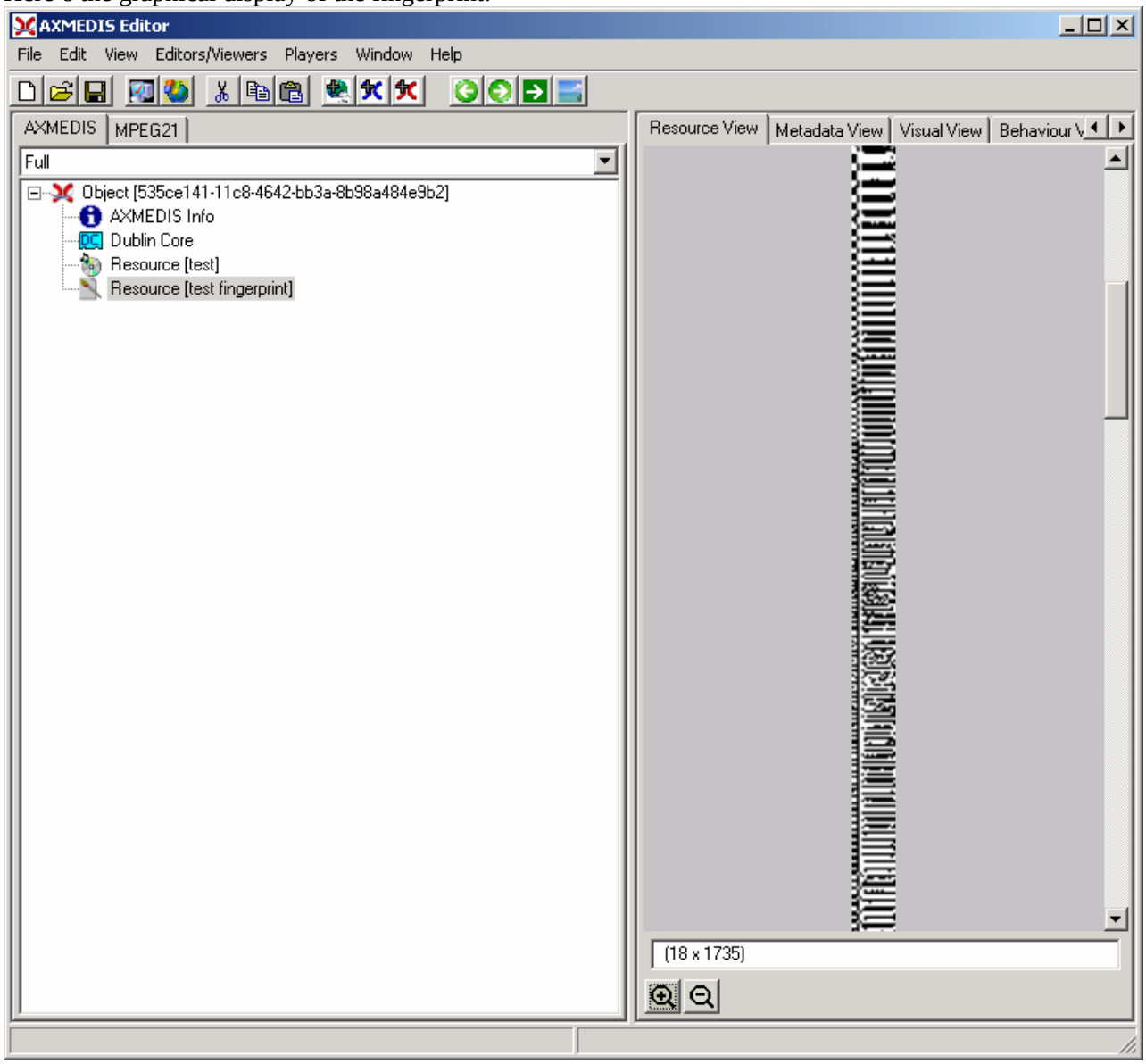


Right Click on the resource and select ‘Content Processing Plug-ins’, Select ‘AudioFingerprintExtraction’ and the following window should appear:



After receiving the ‘success’ message, close the window and you should have two new resources in the AXMEDIS editor. One binary (“fingerprint/audio”) and a graphical one (“image/x-ms-bmp”).

Here's the graphical display of the fingerprint:



More detailed information is available in the plug-in description.

17.2.6 Errors reported and that may occur

Error code
ERROR: Audiofile too short to extract fingerprint
ERROR: Error calculating the power in the Bands
ERROR: Input stream not found or could not open input stream!
ERROR: Could not read from Input Stream. Filesize was corrupted!
ERROR: Not enough free memory in system! Ran out of memory!
ERROR: Error opening file!
ERROR: Couldn't find stream information!
ERROR: Didn't find an audio stream!
ERROR: Ran out of memory! unable to allocate block of memory for the samples!
ERROR: Codec not installed!
ERROR: Could not open Codec!
ERROR: Error while decoding frame!
ERROR: Client provided a corrupted Input Stream!
ERROR: Could not read fingerprint file! Verfy source
ERROR: No data to copy!

17.2.7 Formal description of algorithm

AudioFingerprinting	
Method	AudioFingerprinting
Description	Calculated the perceptual hash value for audio
Input parameters	InputResource: audio file (or video file for which the audio stream is analysed) nFeatures: number of features to be considered frameSize: window size for which a subfingerprint is calculated frameShift: window shift (determines the overlaps) offset: offset information
Output parameters	OutputResource: binary vector and image

18 Fingerprint Estimation for Video files (FHGIGD)

18.1 FIPSVideo

Module/Tool Profile		
FIPSVideo		
Responsible Name	Martin Schmucker (FHGIGD)	
Responsible Partner	FHGIGD	
Status (proposed/approved)	Proposed	
Implemented/not implemented	Implemented	
Status of the implementation	100%	
Executable or Library/module (Support)	Library (PlugIn)	
Single Thread or Multithread	Multithreaded	
Language of Development	C++	
Platforms supported	Microsoft Windows 32	
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/newrepos/Framework/include/fingerprint/video https://cvs.axmedis.org/newrepos/Framework/source/fingerprint/video https://cvs.axmedis.org/newrepos/Framework/project/fingerprint/video	
Reference to the AXFW location of the demonstrator executable tool for internal download	https://cvs. axmedis.org/newrepos/Framework/bin/fingerprint/video	
Reference to the AXFW location of the demonstrator executable tool for public download		
Address for accessing to WebServices if any, add accession information (user and Passwd) if any		
Test cases (present/absent)	absent	
Test cases location	-	
Usage of the AXMEDIS configuration manager (yes/no)		
Usage of the AXMEDIS Error Manager (yes/no)		
Major Problems not solved	-- --	
Major pending requirements	-- --	
Interfaces API with other tools, named as	Name of the communicating tools References to other major components needed	Communication model and format (protected or not, etc.)
Formats Used	Shared with	format name or reference to a section
Protocol Used	Shared with	Protocol name or reference to a section

Used Database name		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
FFMPEG		
ImageMagick		

18.1.1 General Description of the Module

Video files are related to audio as video also is time dependent: They are streams. The input stream is segmented. For each segment a “sub-fingerprint” is calculated. Depending on the length of the input sequence, typically a request to the database does not result in a single fingerprint but in an array of fingerprints.

The characteristic processing steps the same as for audio

Feature extraction and processing: The input signal is pre-processed, which depends on the data type. Typical pre-processing operations include resizing or colour conversion. The input data are segmented and so-called "sub-fingerprints" are calculated. Features are generally extracted from a transformation domain. This transformation domain redundancy is decreased (similar to compression). Within this transformation domain relevant features are extracted. In a post-processing specific relative measures can be derived.

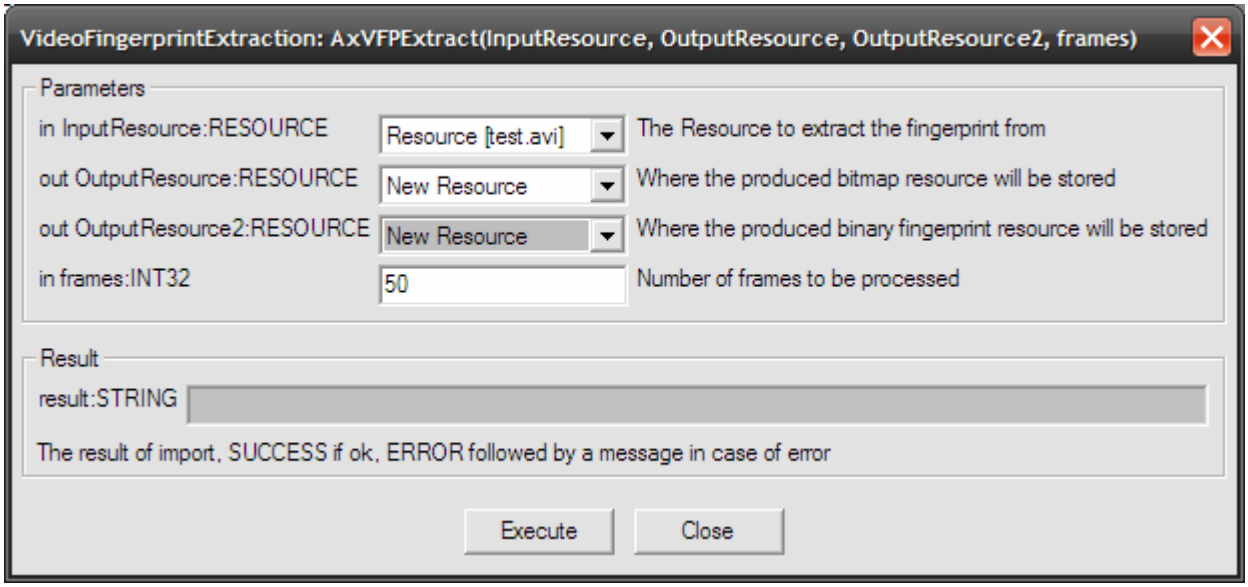
Fingerprint modelling: The multi-dimensional input vector sequence is mapped to a single vector to produce compact fingerprints. This can also include a binarization.

18.1.2 Module Design in terms of Classes

The main integration work consists of the implementation class `VideoFingerprinting`, which calls the functionality developed within AXMEDIS. This library is a wrapper class that calls the corresponding functionalities of the FFMPEG library.

18.1.3 User interface description

Usage of the developed plug-in depends on the AXMEDIS program that utilizes the available functionality. As shown in the next figure – which is an example when the plug-in is used in the AXEditor – the only parameter for the fingerprint calculation is the number of the considered frames.



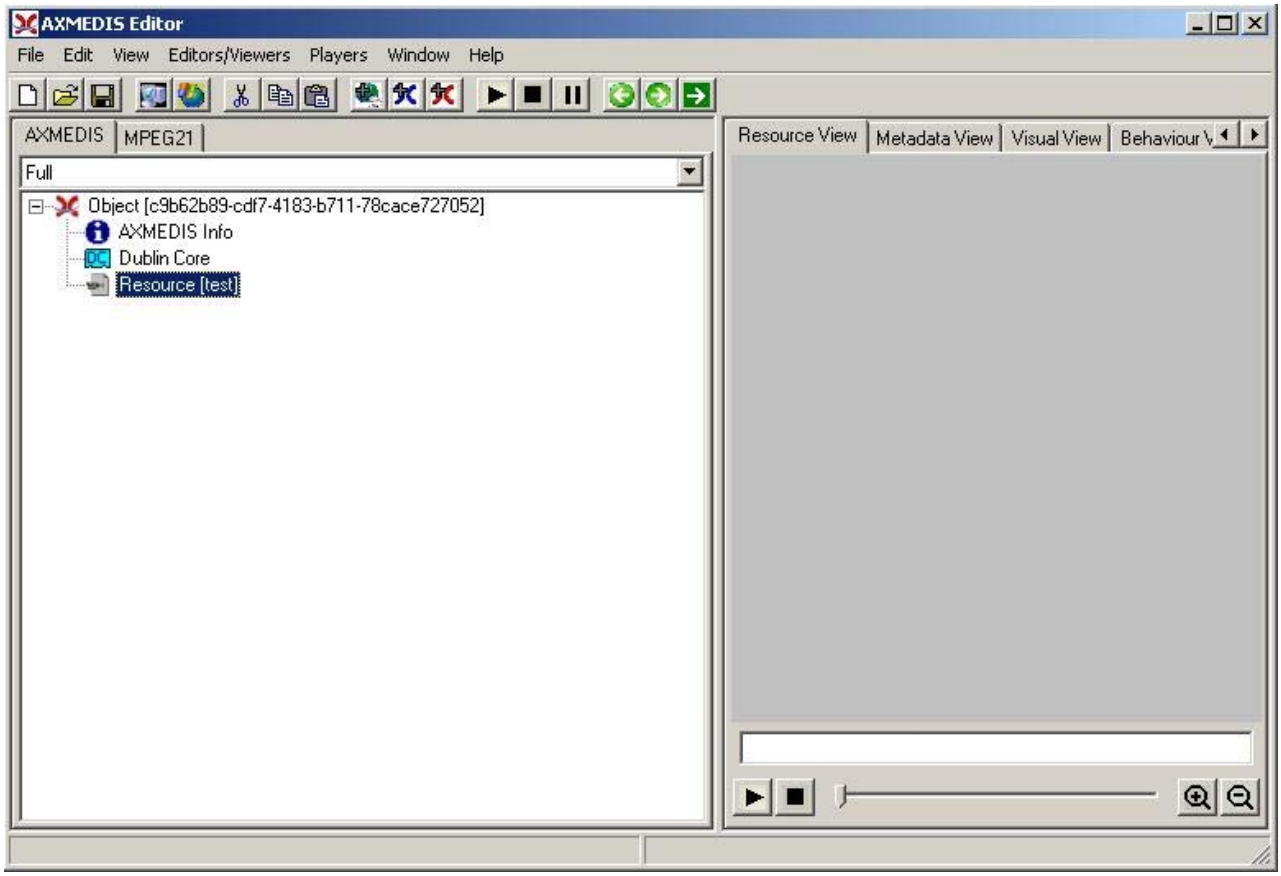
18.1.4 Technical and Installation information

References to other major components needed	Dynamic Link Libraries (DLLs) of the FFMPEG library and ImageMagick for the visual output (avcodec-51_AXFP.dll, avformat-50_AXFP.dll and avutil-49_AXFP.dll).	Dynamic Lin and avformat.
Problems not solved	-	-
Configuration and execution context	Plug-ins are installed by copying the library and the library description in the corresponding plug-in directory.	-

18.1.5 Draft User Manual and Examples of usage

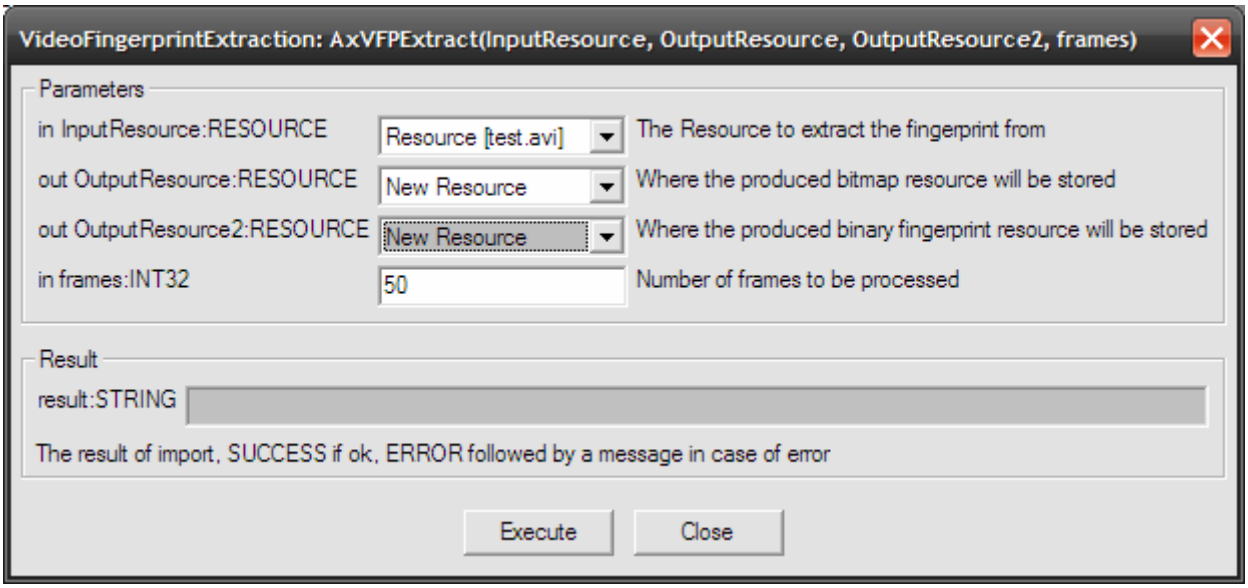
The plug-in can be applied to any video resources, provided this was declared in the mime type attribute of the resource. The output is of mimetype “image/x-ms-bmp”.

Create a new AXMEDIS object and with a right click, add the avi file as an embedded resource.

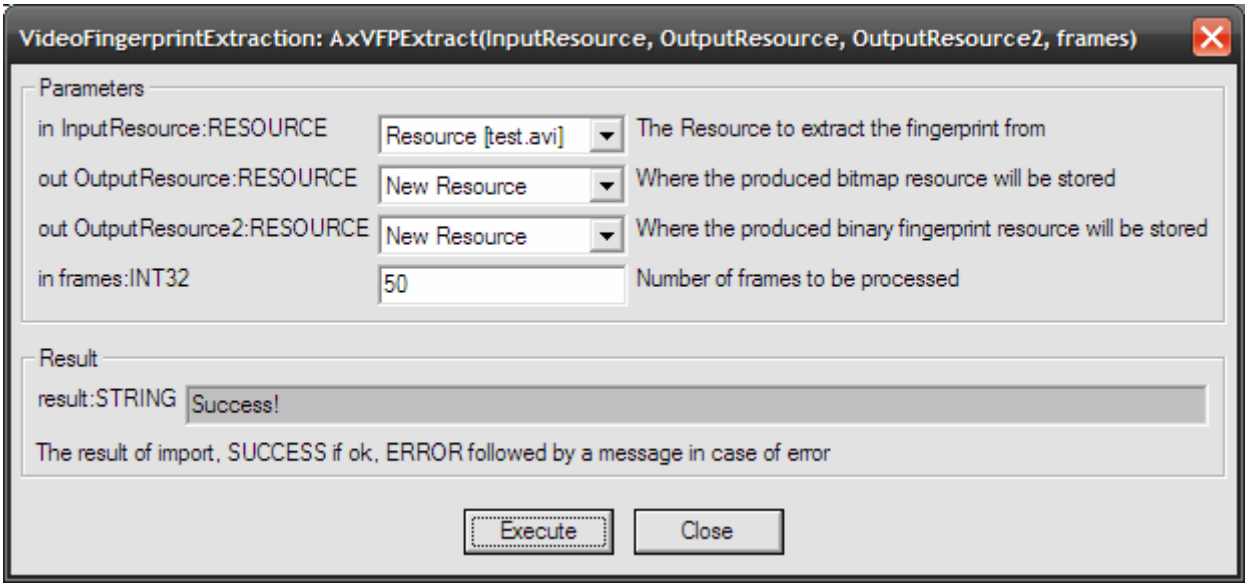


With a right Click on the resource, select ‘Content Processing Plug-ins’. Then you can search for the option ‘VideoFingerprintExtraction’ and click execute.

After the selection, the following window should appear:

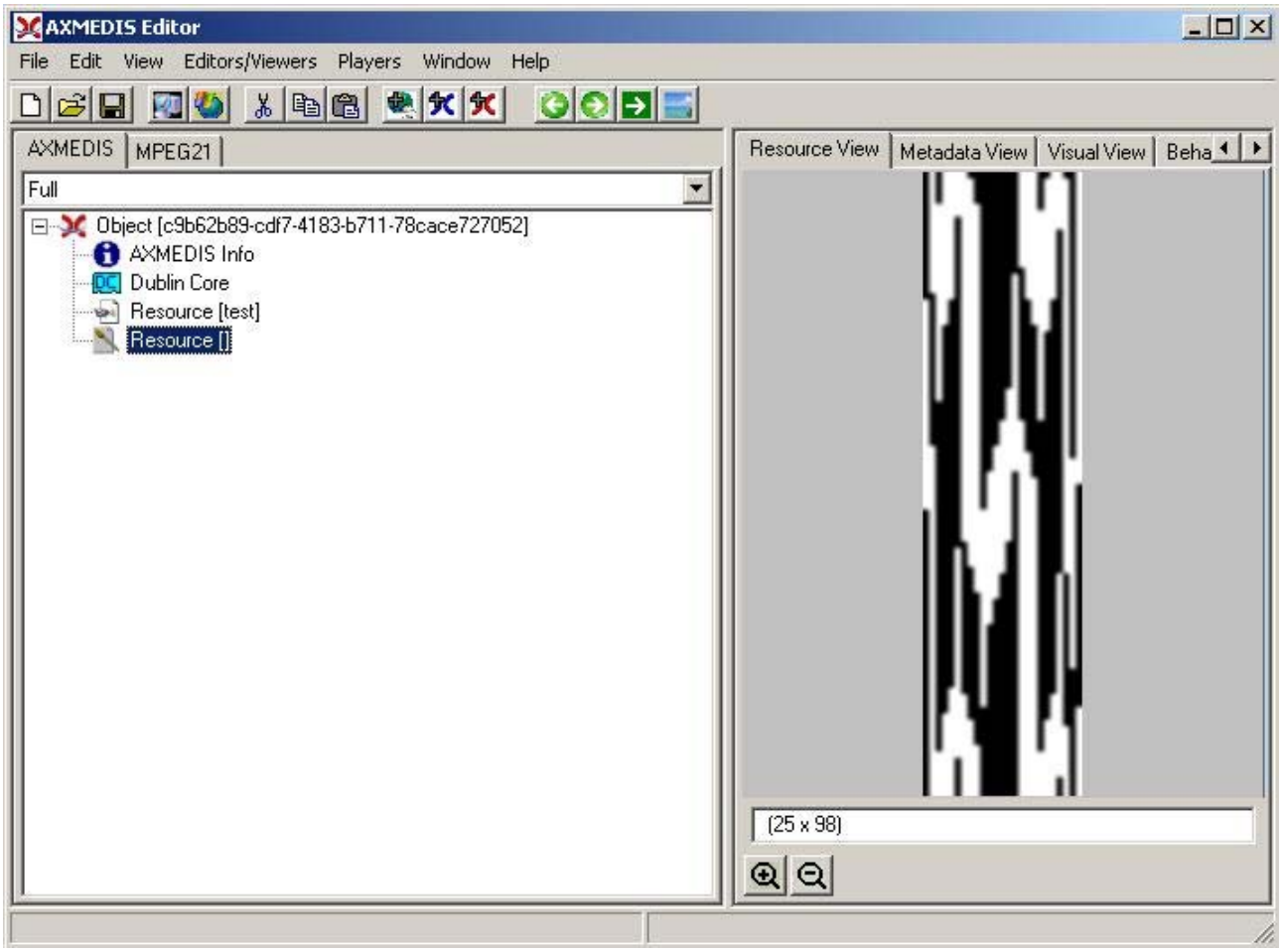


Make outputs as new resources, select the desired number of frames to be processed and click execute.



After receiving the 'success' message, close this window. You should have two new resources in the AXMEDIS editor

With a double click on the image resource, the following graphical of the fingerprint will be showed:



18.1.6 Errors reported and that may occur

Error code
`ERROR: Could not read fingerprint file! Verify source`

18.1.7 Formal description of algorithm

VideoFingerprinting	
Method	VideoFingerprinting
Description	Calculates the perceptual hash for a video file
Input parameters	InputResource: input video nFrames: number of frames to be considered
Output parameters	OutputResource: binary vector and image

19 Fingerprint Estimation for Metadata (FHGIGD)

19.1 FIPSMetaData

Module/Tool Profile		
FIPSMetaData		
Responsible Name	Martin Schmucker (FHGIGD)	
Responsible Partner	FHGIGD	
Status (proposed/approved)	Proposed	
Implemented/not implemented	Implemented	
Status of the implementation	100%	
Executable or Library/module (Support)	Library (PlugIn)	
Single Thread or Multithread	Multithreaded	
Language of Development	C++	
Platforms supported	Microsoft Windows 32	
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/newrepos/Framework/include/fingerprint/metadata https://cvs.axmedis.org/newrepos/Framework/source/fingerprint/metadata https://cvs.axmedis.org/newrepos/Framework/project/fingerprint/metadata	
Reference to the AXFW location of the demonstrator executable tool for internal download	https://cvs.axmedis.org/newrepos/Framework/bin/fingerprint/metadata	
Reference to the AXFW location of the demonstrator executable tool for public download		
Address for accessing to WebServices if any, add accession information (user and Passwd) if any		
Test cases (present/absent)	Absent	
Test cases location	-	
Usage of the AXMEDIS configuration manager (yes/no)		
Usage of the AXMEDIS Error Manager (yes/no)		
Major Problems not solved	-- --	
Major pending requirements	-- --	
Interfaces API with other tools, named as	Name of the communicating tools References to other major components needed	Communication model and format (protected or not, etc.)

Formats Used	Shared with	format name or reference to a section
Protocol Used	Shared with	Protocol name or reference to a section
Used Database name		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not

19.1.1 General Description of the Module

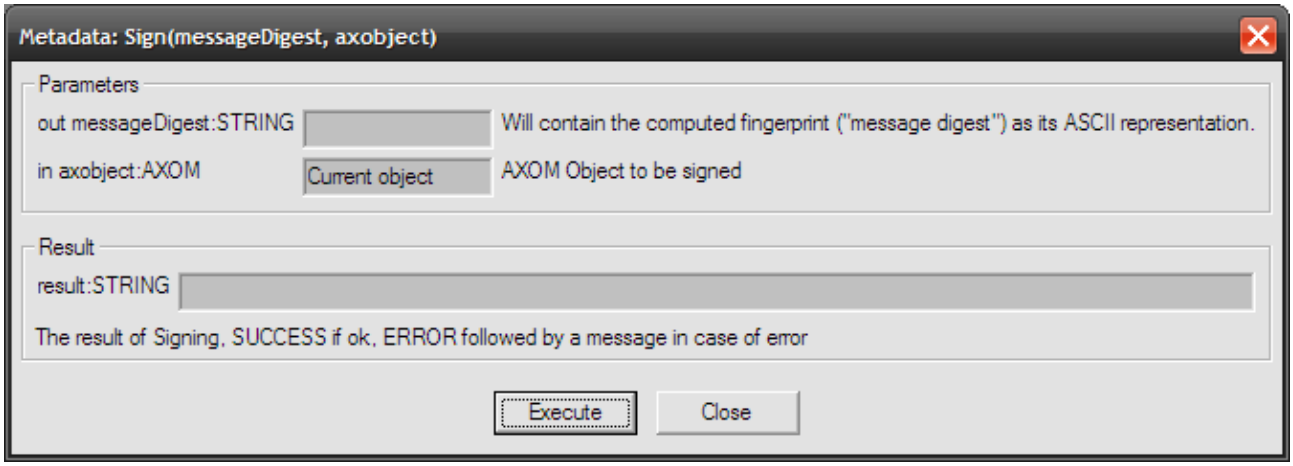
For the verification of the objects metadata only cryptographic hash functions are feasible. This hash function is applied to all kinds of meta-data of an AXMEDIS object.

19.1.2 Module Design in terms of Classes

The main integration work consists of the integration of the class MetaData `Fingerprinting`, which calls the functionality implemented within AXMEDIS.

19.1.3 User interface description

Usage of the developed plug-in depends on the AXMEDIS program that utilizes the available functionality. As shown in the next figure – which is an example when the plug-in is used in the AXEditor – no parameters has to be set.

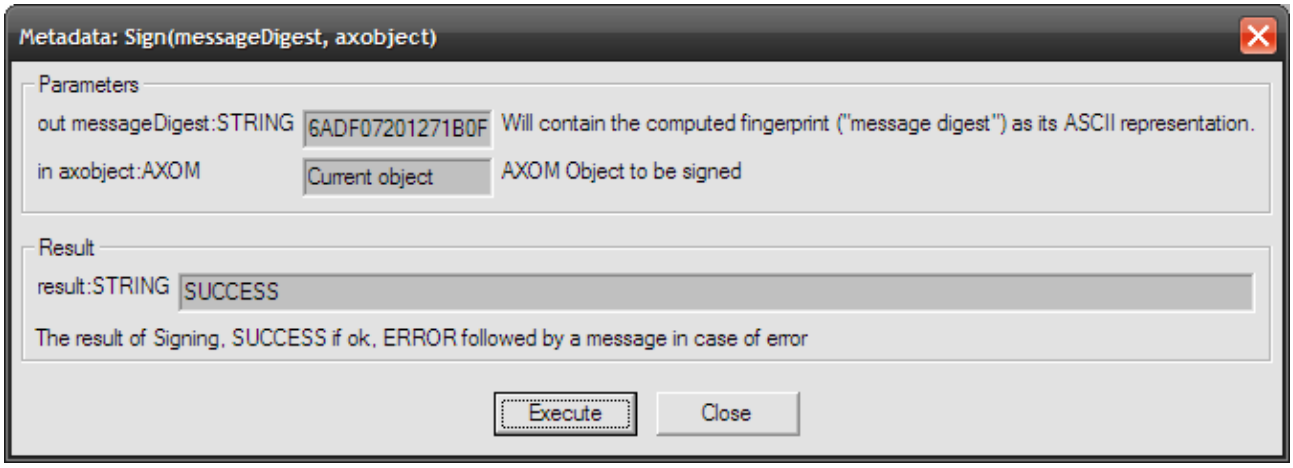


19.1.4 Technical and Installation information

References to other major components needed	-
Problems not solved	-
Configuration and execution context	Plug-ins are installed by copying the library and the library description in the corresponding plug-in directory.

Draft User Manual and Examples of usage

Select a MetaData object and select via the right mouse button the functionality to calculate the cryptographic hash value for the selected meta data. The result will be shown and given back as a string value:



19.1.5 Errors reported and that may occur

Error code
`ERROR: cannot completely process the resource`
`ERROR: cannot convert fingerprint to a string`

19.1.6 Formal description of algorithm

MetaDataFingerprinting	
Method	MetaDataFingerprinting
Description	Calculated the cryptographic value for any meta data

Input parameters	InputResource: XML MetaData-Description
Output parameters	Output: StringValue

20 Fingerprint Estimation for Generic Files (FHGIGD)

20.1 FIPSMetaData

Module/Tool Profile		
FIPSMetaData		
Responsible Name	Martin Schmucker (FHGIGD)	
Responsible Partner	FHGIGD	
Status (proposed/approved)	Proposed	
Implemented/not implemented	Implemented	
Status of the implementation	100%	
Executable or Library/module (Support)	Library (PlugIn)	
Single Thread or Multithread	Multithreaded	
Language of Development	C++	
Platforms supported	Microsoft Windows 32	
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/newrepos/Framework/include/fingerprint/general_resource https://cvs.axmedis.org/newrepos/Framework/source/fingerprint/general_resource https://cvs.axmedis.org/newrepos/Framework/project/fingerprint/general_resource	
Reference to the AXFW location of the demonstrator executable tool for internal download	https://cvs.axmedis.org/newrepos/Framework/bin/fingerprint/general_resource	
Reference to the AXFW location of the demonstrator executable tool for public download		
Address for accessing to WebServices if any, add accession information (user and Passwd) if any		
Test cases (present/absent)	Absent	
Test cases location	-	
Usage of the AXMEDIS configuration manager (yes/no)		
Usage of the AXMEDIS Error Manager (yes/no)		
Major Problems not solved	-- --	
Major pending requirements	-- --	
Interfaces API with other tools, named as	Name of the communicating tools References to other major components needed	Communication model and format (protected or not, etc.)

Formats Used	Shared with	format name or reference to a section
Protocol Used	Shared with	Protocol name or reference to a section
Used Database name		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not

20.1.1 General Description of the Module

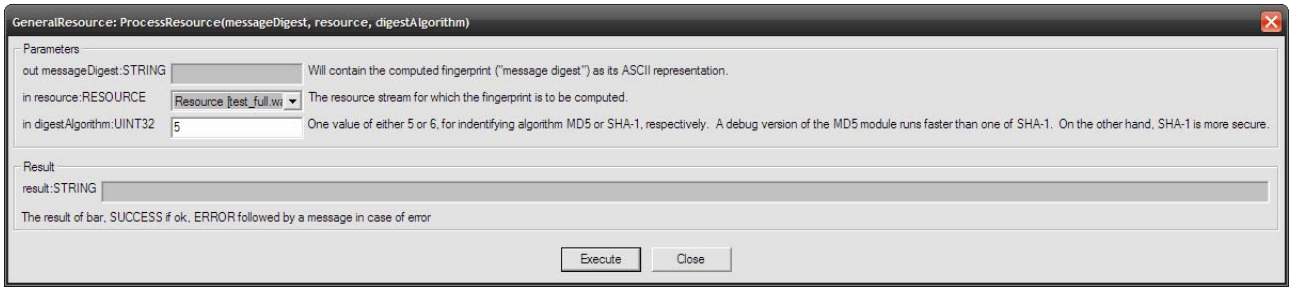
For the verification of the objects metadata only cryptographic hash functions are feasible. This hash function is applied to all kinds of meta-data of an AXMEDIS object.

20.1.2 Module Design in terms of Classes

The main integration work consists of the integration of the class General Resource `Fingerprinting`, which calls the functionality implemented within AXMEDIS.

20.1.3 User interface description

Usage of the developed plug-in depends on the AXMEDIS program that utilizes the available functionality. As shown in the next figure – which is an example when the plug-in is used in the AXEditor – several parameters can be set.



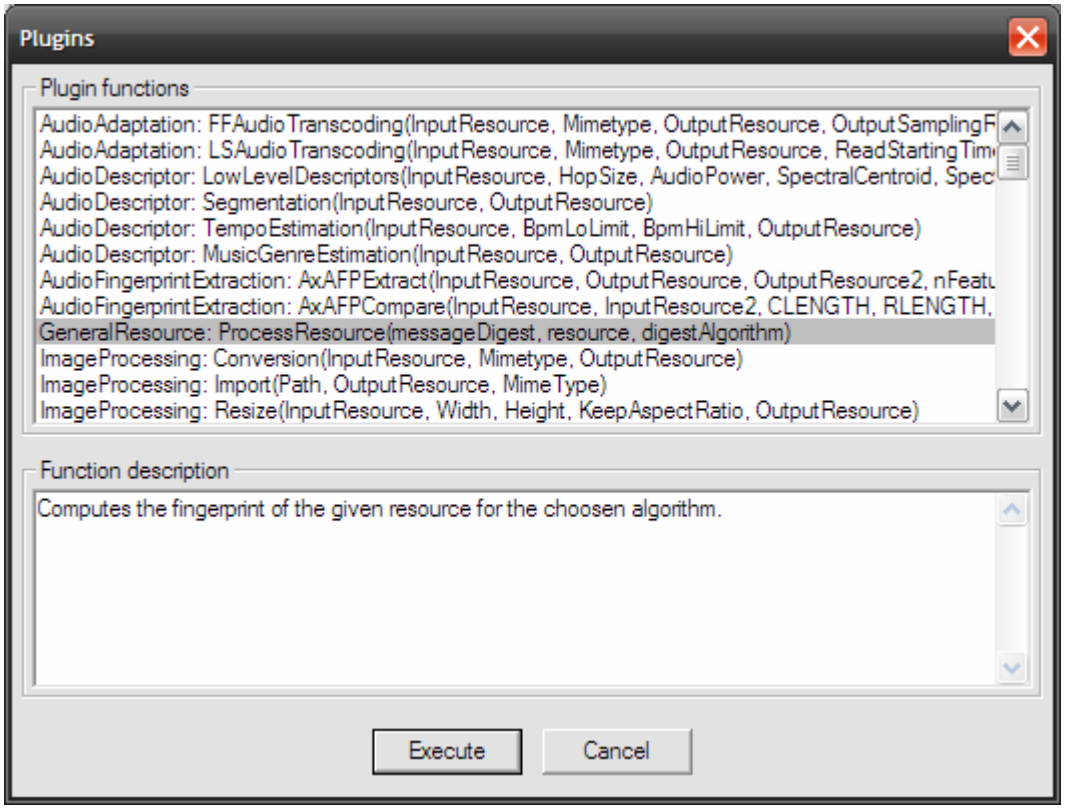
20.1.4 Technical and Installation information

References to other major components needed	-
Problems not solved	-
Configuration and execution context	Plug-ins are installed by copying the library and the library description in the corresponding plug-in directory.

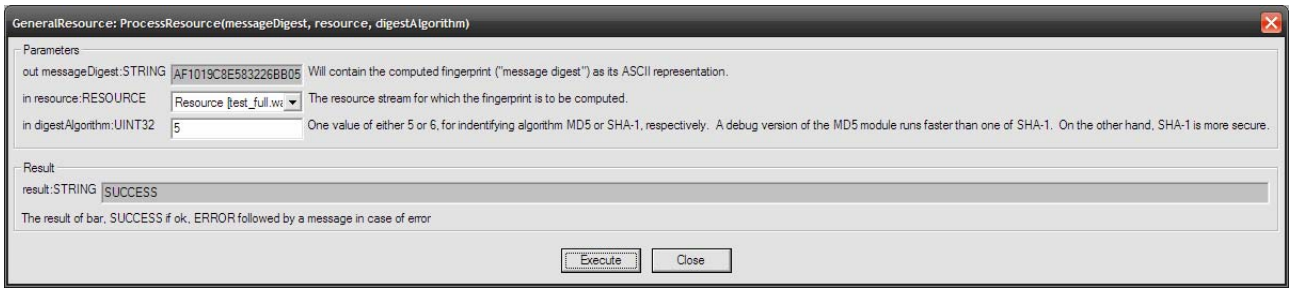
20.1.5 Draft User Manual and Examples of usage

The plug-in can be applied to any content type. The output is a string.

Create a new AXMEDIS object and a resource or open one object with embedded resource(s). On the AXMEDIS object chose “Content Processing Plug-in…” and select “GeneralResource”.



Select the available algorithm (5: MD5, 6: SHA-1) and press “Execute”.



20.1.6 Errors reported and that may occur

Error code

```
ERROR: cannot completely process the resource
ERROR: cannot convert fingerprint to a string
```

Formal description of algorithm

GenericFileFingerprinting	
Method	GenericFileFingerprinting
Description	Calculated the cryptographic value for any resource
Input parameters	InputResource: XML MetaData-Description
Output parameters	Output: StringValue

21 Fingerprint Formats (FHGIGD)

As for the descriptor formats, the fingerprint formats are based on the corresponding MPEG standardization. Metadate and corresponding descriptors are addressed in MPEG-7. Again, this section briefly describes the corresponding MPEG-7 descriptors where they are suitable. If they are not suitable, an explanation is given.

21.1 Text Fingerprints defined in MPEG-7 (DIPITA)

In MPEG-7 this kind of meta-information (for verification of the text documents) is not consider so far. Within AXMEDIS it has to be identified, how cryptographic hash values can be integrated best into MPEG-7 and the relationship with MPEG-21.

21.2 Audio Fingerprints defined in MPEG-7 (FHGIGD):

Within MPEG-7 low-level descriptors for audio are already defined. Below the main descriptors of MPEG-7 for the fingerprinting of audio file are described:

```
<!-- ################################################################ -->
<!-- Definition of AudioLLDScalarType                                 -->
<!-- ################################################################ -->
<complexType name="AudioLLDScalarType" abstract="true">
   <complexContent>
      <extension base="mpeg7:AudioDType">
         <choice>
            <element name="Scalar" type="float"/>
            <element name="SeriesOfScalar">
               <complexType>
                  <complexContent>
                     <extension base="mpeg7:SeriesOfScalarType">
                        <attribute      name="hopSize"      type="mpeg7:mediaDurationType"
use="optional" default="10F1000"/>
                     </extension>
                  </complexContent>
               </complexType>
            </element>
         </choice>
      </extension>
   </complexContent>
</complexType>
```

Name	***Definition***
`AudioLLDScalarType`	Abstract definition inherited by all scalar datatype audio descriptors.
`Scalar`	Value of the descriptor
`SeriesOfScalar`	Scalar values for sampled-series description of an audio segment. Use of this scalable series datatype promotes compatibility between sampled descriptions.
`hopSize`	Time interval between data samples for series description. The default value is `PT10N1000F` which is 10 milliseconds. Values other than the default shall be integer multiples/divisors of 10 milliseconds. This will ensure compatibility of descriptors sampled at different rates.

```
<!-- ################################################################ -->
<!-- Definition of AudioLLDVectorType                                 -->
<!-- ################################################################ -->
<complexType name="AudioLLDVectorType" abstract="true">
   <complexContent>
      <extension base="mpeg7:AudioDType">
         <choice>
            <element name="Vector" type="mpeg7:floatVector"/>
```

```
            <element name="SeriesOfVector">
               <complexType>
                  <complexContent>
                     <extension base="mpeg7:SeriesOfVectorType">
                        <attribute      name="hopSize"      type="mpeg7:mediaDurationType"
use="optional" default="10F1000"/>
                     </extension>
                  </complexContent>
               </complexType>
            </element>
         </choice>
      </extension>
   </complexContent>
</complexType>
```

Name	Definition
AudioLLDVectorType	Abstract definition inherited by all vector datatype audio descriptors.
Vector	Vector value of descriptor
SeriesOfVector	Vector values for sampled-series description of an audio segment. Use of this scalable series datatype promotes compatibility between sampled descriptions.
hopSize	Time interval between data samples for series description. The default value is PT10N1000F which is 10 milliseconds. Values other than the default shall be integer multiples/divisors of 10 milliseconds. This will ensure compatibility of descriptors sampled at different rates.

21.3 Video Fingerprints defined in MPEG-7 (FHGIGD):

As described before, MPEG-7 defines data type for regular and irregular visual time series depending on the (non-) constant intervals between succeeding descriptors.

```
<!-- ################################################ -->
<!--   Definition of the TimeSeriesDatatype        -->
<!-- ################################################ -->
<complexType name="TimeSeriesType" abstract="true">
   <sequence>
      <element name="TimeIncr" type="mpeg7:mediaDurationType"/>
   </sequence>
   <attribute      name="offset"      type="mpeg7:mediaDurationType"      use="optional"
default="PT0S"/>
</complexType>

<!-- ################################################ -->
<!--   Definition of the VisualTimeSeriesDatatype        -->
<!-- ################################################ -->
<complexType name="VisualTimeSeriesType" abstract="true">
   <sequence>
      <element name="TimeIncr" type="mpeg7:mediaDurationType"/>
   </sequence>
   <attribute      name="offset"      type="mpeg7:mediaDurationType"      use="optional"
default="PT0S"/>
</complexType>

<!--   Definition of the RegularTimeSeries Datatype        -->
<complexType name="RegularTimeSeriesType" final="#all">
   <complexContent>
      <extension base="mpeg7:VisualTimeSeriesType">
         <sequence>
            <element     name="Descriptor"     type="mpeg7:VisualDType"     minOccurs="1"
maxOccurs="unbounded"/>
         </sequence>
      </extension>
```

```
   </complexContent>
</complexType>

<!--   Definition of the IrregularTimeSeries Datatype        -->
<complexType name="IrregularTimeSeriesType" final="#all">
   <complexContent>
      <extension base="mpeg7:VisualTimeSeriesType">
         <sequence minOccurs="1" maxOccurs="unbounded">
            <element name="Descriptor" type="mpeg7:VisualDType"/>
            <element name="Interval" type="mpeg7:unsigned32"/>
         </sequence>
      </extension>
   </complexContent>
</complexType>
```

As no low level descriptor data type is defined for video data, which can store video fingerprints, a descriptor has to be defined within AXMEDIS. This is related to the existing ColorStructureType:

```
<complexType name=″ColorStructureType″ final=″#all″>
   <complexContent>
      <extension base=″mpeg7:VisualDType″>
         <sequence>
            <element name=″Values″>
               <simpleType>
                  <restriction>
                     <simpleType>
                        <list itemType=″mpeg7:unsigned8″/>
                     </simpleType>
                     <minLength value=″1″/>
                     <maxLength value=″256″/>
                  </restriction>
               </simpleType>
            </element>
         </sequence>
         <attribute name=″colorQuant″ type=″mpeg7:unsigned3″ use=″required″/>
      </extension>
   </complexContent>
</complexType>
```

However, instead of describing the colour structure a “VideoLLDScalar” and a “VideoLLDVectorType” are proposed, which have to be further evaluated according to the needs within AXMEDIS and general needs:

```
<!-- ################################################################################ -->
<!-- Definition of VideoLLDScalarType                                                 -->
<!-- ################################################################################ -->
<complexType name="VideoLLDScalarType" abstract="true">
   <complexContent>
      <extension base="mpeg7:VisualDType">
         <choice>
            <element name="Scalar" type="float"/>
            <element name="SeriesOfScalar">
               <complexType>
                  <complexContent>
                     <extension base="mpeg7:SeriesOfScalarType">
                        <attribute      name="hopSize"      type="mpeg7:mediaDurationType"
use="optional" default="10F1000"/>
                     </extension>
                  </complexContent>
               </complexType>
            </element>
         </choice>
      </extension>
   </complexContent>
</complexType>
```

```
<!-- ################################################################### -->
<!-- Definition of VideoLLDVectorType                                    -->
<!-- ################################################################### -->
<complexType name="VideoLLDVectorType" abstract="true">
   <complexContent>
      <extension base="mpeg7: VisualDType ">
         <choice>
            <element name="Vector" type="mpeg7:floatVector"/>
            <element name="SeriesOfVector">
               <complexType>
                  <complexContent>
                     <extension base="mpeg7:SeriesOfVectorType">
                        <attribute      name="hopSize"      type="mpeg7:mediaDurationType"
use="optional" default="10F1000"/>
                     </extension>
                  </complexContent>
               </complexType>
            </element>
         </choice>
      </extension>
   </complexContent>
</complexType>
```

21.4 Fingerprint Extractors for Any Digital Files (FHGIGD)

For the verification of the objects metadata only cryptographic hash functions are feasible. This hash function is applied to all kinds of meta-data of an AXMEDIS object.

In MPEG-7 this kind of meta-information (for verification of the meta-data) is not consider so far. Ideally, each description should contain a cryptographic hash function to ensure the link between the object and meta-data. This is not yet foreseen within MPEG-7:

```
<!-- ################################################ -->
<!--  Definition of Mpeg7RootType                     -->
<!-- ################################################ -->

<complexType name="Mpeg7RootType" abstract="true">
   <attribute ref="xml:lang" use="optional"/>
   <attribute name="name" type="NCName" use="optional"/>
   <attribute name="version" type="string" use="optional"/>
   <attribute name="creationTime" type="mpeg7:timePointType" use="optional"/>
   <attribute name="creationPlace" type="string" use="optional"/>
   <attribute name="creationTool" type="string" use="optional"/>
   <attribute name="creator" type="string" use="optional"/>
   <attribute name="owner" type="string" use="optional"/>
   <attribute name="copyright" type="string" use="optional"/>
</complexType>
```

Within AXMEDIS it has to be identified, how cryptographic hash values can be integrated best into MPEG-7 and the relationship with MPEG-21.

22 External Protection Libraries (EPFL)

22.1.1 Cryptography tools and algorithms for security processing

Module/Tool Profile Cryptography Tool and Algorithms (Cryptlib)		
Responsible Name	Mattavelli	
Responsible Partner	EPFL	
Status (proposed/approved)	Proposed	
Implemented/not implemented	Implemented	
Status of the implementation	Plug-In	
Executable or Library/module (Support)	Library	
Single Thread or Multithread	Multithread	
Language of Development	C	
Platforms supported	Win32, Linux, Unix	
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/newrepos/Software/Applications/Cryptlib/Crypt	
Reference to the AXFW location of the demonstrator executable tool for internal download	https://cvs.axmedis.org/newrepos/Software/Applications/Cryptlib/binaries	
Reference to the AXFW location of the demonstrator executable tool for public download		
Address for accessing to WebServices if any, add accession information (user and Passwd) if any		
Test cases (present/absent)	Present	
Test cases location	https://cvs.axmedis.org/newrepos/Software/Applications/Cryptlib/Binaries	
Usage of the AXMEDIS configuration manager (yes/no)	No	
Usage of the AXMEDIS Error Manager (yes/no)	No	
Major Problems not solved	-- --	
Major pending requirements	-- --	
Interfaces API with other tools, named as	Name of the communicating tools References to other major components needed	Communication model and format (protected or not, etc.)
Formats Used	Shared with	format name or reference to a section

Protocol Used	Shared with	Protocol name or reference to a section
Used Database name		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not

22.1.2 General Description of the Module

The information age has seen the development of electronic pathways that carry vast amounts of valuable commercial content between individuals and companies.
Unfortunately the unprecedented levels of access provided by systems like the Internet also expose this data to breaches of confidentiality, disruption of service, and copyrights infringements.
For this reason, in the content distribution field, many applications use more and more DRM (Digital rights Managements) solutions.
This means that content use/manage applications need security module implemented within.
Unfortunately the security systems required to protect data are generally extremely difficult to design and implement, and even when available tend to require considerable understanding of the underlying principles in order to be used. This has lead to a proliferation of “snake oil” products that offer only illusionary security, or to organizations holding back from deploying online information systems because the means to secure them are not readily available, or because they employed weak, easily broken security that was unacceptable to users.
The cryptlib security library provides a complete set of cryptographic algorithms that fit the Axmedis needs.
The following subsections describe the external library that may be used in the AXMEDIS framework to implement the needed cryptography functionalities.

22.1.3 User interface description

The cryptlib functionalities could be used as plug-ins through the AXCP interface. The interface of AXCP plug-ins can map the formal description of the function and allows entering textually all parameters of the function (key, Mode, Algorithm).

22.1.4 Technical and Installation information

The cryptographic functionalities will be used as plug-ins through the AXCP interface. The plug-in simply consists of a DLL and an XML file describing the functionalities of the DLL. Both the DLL and the XML description should be installed in the plug-in directory of the AXCP compliant tool using the plug-in.

References to other major components needed	
Problems not solved	
Configuration and execution context	

22.1.5 Draft User Manual

The integration in the Axmedis framework has to be done.

22.1.6 Integration and compilation issues

The library has been compiled and tested successfully on Win32 platform. It should be ported easily on Linux and Unix platforms though it has not been tested yet.

22.1.7 Configuration Parameters

Config parameter	Possible values
CRYPT_CTXINFO_ALGO CRYPT_CTXINFO_MODE	Algorithm and mode (see sections below)
CRYPT_CTXINFO_BLOCKSIZE	Cipher block size in bytes
CRYPT_CTXINFO_IVSIZE	Cipher IV size in bytes
CRYPT_CTXINFO_KEYING_-ALGO CRYPT_CTXINFO_KEYING_-ITERATIONS CRYPT_CTXINFO_KEYING_-SALT	The algorithm and number of iterations used to transform a user-supplied key or password into an algorithm-specific key for the context, and the salt value used in the transformation process
CRYPT_CTXINFO_KEYSIZE	Key size in bytes
CRYPT_CTXINFO_LABEL	Key label
CRYPT_CTXINFO_NAME_ALGO CRYPT_CTXINFO_NAME_MODE	Algorithm and mode name (see following Section)

22.1.8 Algorithms

This section describes the characteristics of each algorithm used in cryptlib and any known restrictions on their use.

AES AES is a 128-bit block cipher with a 128-bit key and has the cryptlib algorithm identifier CRYPT_ALGO_AES.

Blowfish Blowfish is a 64-bit block cipher with a 448-bit key and has the cryptlib algorithm identifier CRYPT_ALGO_BLOWFISH.

CAST-128 CAST-128 is a 64-bit block cipher with a 128-bit key and has the cryptlib algorithm identifier CRYPT_ALGO_CAST.

DES DES is a 64-bit block cipher with a 56-bit key and has the cryptlib algorithm identifier CRYPT_ALGO_DES. Note that this algorithm is no longer considered secure and should not be used. It is present in cryptlib only for compatibility with legacy applications. Although cryptlib uses 64-bit DES keys, only 56 bits of the key are actually used.

Triple DES Triple DES is a 64-bit block cipher with a 112/168-bit key and has the cryptlib algorithm identifier CRYPT_ALGO_3DES. Although cryptlib uses 128, or 192-bit DES keys (depending on whether two- or three-key triple DES is being used), only 112 or 168 bits of the key are actually used.

Diffie-Hellman Diffie-Hellman is a key exchange algorithm with a key size of up to 4096 bits and has the cryptlib algorithm identifier CRYPT_ALGO_DH. Diffie-Hellman was formerly covered by a patent in the US, this has now expired.

22.1.9 Mode

A a symmetric key algorithm encrypts plaintext in fixed-size n-bit blocks (often n = 64/128/256). For messages exceeding n bits, the simplest approach is to partition the message into n-bit blocks and encrypt each separately. This electronic-codebook (ECB) mode has disadvantages in most applications, motivating other methods of employing block ciphers (*modes of operation*) on larger messages.
The four most common modes are ECB, CBC, CFB, and OFB. These are summarized and discussed below.

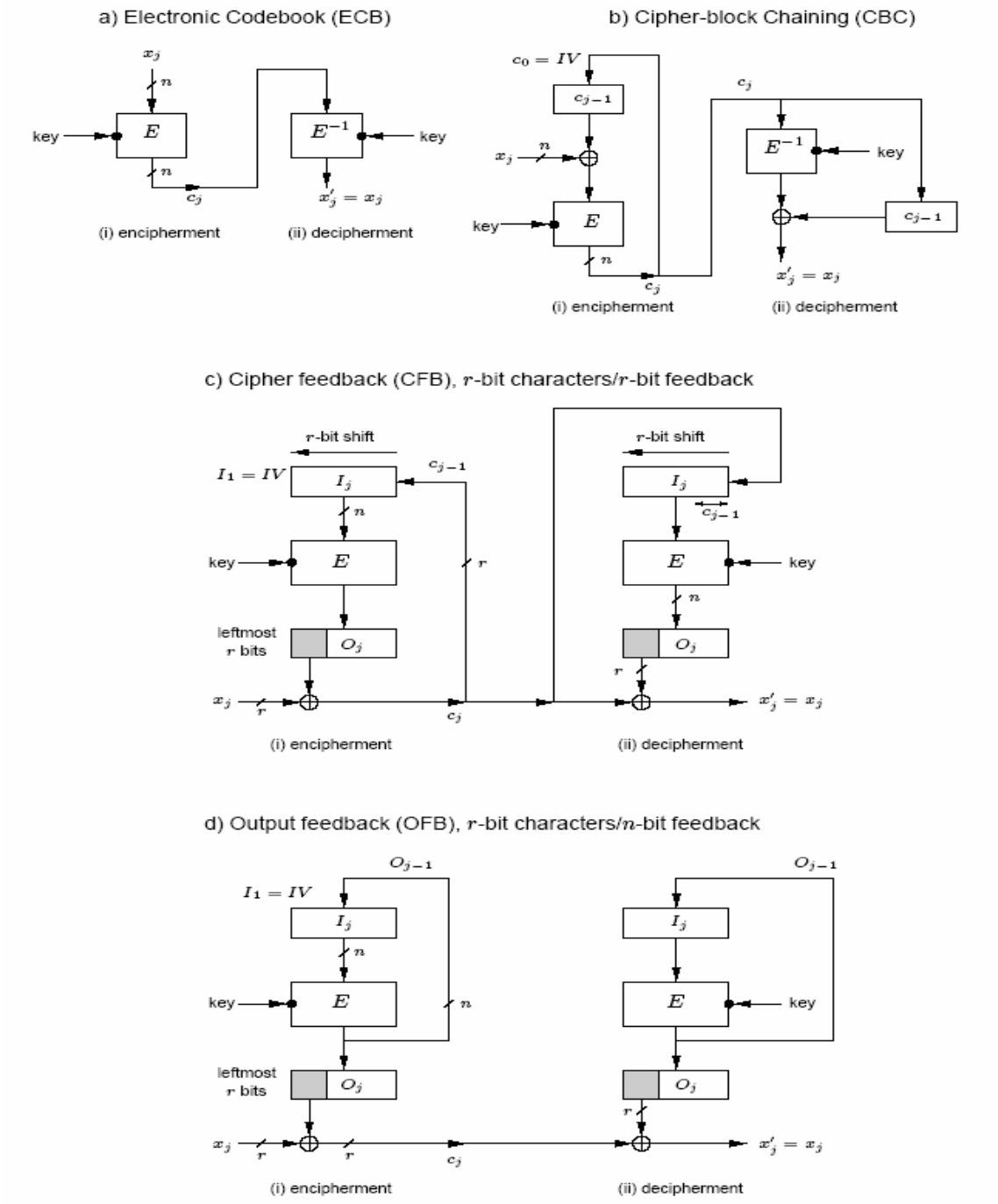
ECB mode The *electronic codebook* (ECB) mode of operation

CBC mode The *cipher-block chaining* (CBC) mode of operation involves use of an n-bit initialization vector, denoted IV

CFB mode the *cipher feedback (CFB)* where some applications require transmission without delay.

OFB mode The *output feedback* (OFB) mode of operation may be used for applications in which all error propagation must be avoided. It is similar to CFB, and allows encryption of various block sizes (characters), but differs in that the output of the encryption block function E (rather than the ciphertext) serves as the feedback.

See figure below.



22.1.10 Formal description of algorithm

Name	
Method	CryptProcess
Description	Encrypts, decrypts, gives hash value and generate key ccording to the imput parmaters
Input parameters	See section algorithms and parameters
Output parameters	Encrypted/Decrypted content or information

23 Appendix: Relevant External Libraries

23.1 FFMPEG

FFMPEG is a complete solution to record, convert and stream audio and video. It is developed under Linux but it can be operated under most operating systems, including Windows.

FFMPEG provides a C API and two libraries:

11. Libavcodec: a library containing all the FFMPEG audio/video encoders and decoders; most codecs were developed from scratch to ensure best performances and high code reusability;
12. Libavformat: a library containing parsers and generators for all common audio/video formats.

FFMPEG is licensed under LGPL. However, it incorporates several modules that are covered under the GPL, notably liba52 (a library for decoding ATSC A/52 streams) and libpostproc (a library for post-processing). If these components are used in a project, then the all project should be distributed under the GPL. Yet, it is possible to avoid linking to these GPL libraries ensuring a full LGPL use of FFMPEG.

Format transcoding is one of the main adaptation functions needed by the AXMEDIS Framework as it implies bitrate reduction when transcoding among compressed formats. In the case of video objects (but this also applies to audio and multimedia objects), the FFMPEG and the FOBS library may be used.

Here is a list of the file formats supported by FFMPEG through the libavformat library; “X” means that encoding (resp. decoding) is supported:

Supported File Format	E**ncoding**	**Decoding**	**Comments**
MPEG audio	X	X	
MPEG 1 systems	X	X	Muxed audio and video
MPEG 2 PS	X	X	Also known as VOB file
MPEG 2 TS		X	Also known as DVB transport stream
ASF	X	X	
AVI	X	X	
WAV	X	X	
Macromedia flash	X	X	Only embedded audio is decoded
FLV	X	X	Macromedia flash video files
Real audio and video	X	X	
Raw AC3	X	X	
Raw MJPEG	X	X	
Raw MPEG video	X	X	
Raw PCM 8/16 bits, mulaw/Alaw	X	X	
Raw CRI ADX audio	X	X	
SUN AU format	X	X	
NUT	X	X	NUT open container format
Quicktime	X	X	
MPEG4	X	X	MPEG4 is a variant of Quicktime
Raw MPEG4 video	X	X	
DV	X	X	
4xm		X	4X Technologies format, used in some games

Playstation STR		X	
Id RoQ		X	Used in Quake III, Jedi Knight II, other computer games
Interplay MVE		X	Format used in various Interplay computer games
WC3 Movie		X	Multimedia format used in Origin's Wing Commnader III computer game
Sega FILM/CPK		X	Used in many Sega Saturn console games
Westwood Studios VQA/AUD		X	Multimedia formats used in Westwood Studios games
Id Cinematic (.cin)		X	Used in Quake II
FLIC format		X	.fli / ,flc files
Sierra VMD		X	Used in Sierra CD-ROM games
Sierra Online		X	.sol files used in Sierra Online games
Matroska		X	
Electronic Arts Multimedia		X	Used in various EA games; files have extensions like WVE and UV2

Furthermore, FFMPEG can read and write images for each frame of a video sequence. The following image formats are supported:

Supported Image Format	**Encoding**	**Decoding**	**Comments**
PGM, PPM	X	X	
PAM	X	X	PAM is a PNM extension with alpha support
PGMYUV	X	X	PGM with U and V components in YUV 4:2:0
JPEG	X	X	Progressive JPEG is not supported
.Y.U.V.	X	X	One raw file per component
Animated GIF	X	X	Only uncompressed GIF are generated
PNG	X	X	2 bit and 4 bit/pixel not supported yet
SGI	X	X	SGI RGB image format

Here is a list of video codecs supported by FFMPEG through the libavcodec library:

Supported Codec	E**ncoding**	**Decoding**	**Comments**
MPEG1 video	X	X	
MPEG2 video	X	X	
MPEG4	X	X	Also known as DIVX 4/5
MSMPEG4 V1	X	X	
MSMPEG4 V2	X	X	
MSMPEG4 V3	X	X	Also known as DIVX 3
WMV7	X	X	
WMV8	X	X	Not completely working
H. 261	X	X	
H. 263 (+)	X	X	Also known as Real Video 1.0
H. 264		X	
MJPEG	X	X	
Lossless MJPEG	X	X	
Apple MJPEG-B		X	
Sunplus MJPEG		X	Fourcc: SP5X
DV	X	X	

Huff YUV	X	X	
FFmpeg video 1	X	X	Experimental lossless codec (fourcc: FFV1)
FFmpeg snow	X	X	Experimental wavelet codec (fourcc: SNOW)
Asus v1	X	X	Fourcc: ASV1
Asus v2	X	X	Fourcc: ASV2
Creative YUV		X	Fourcc: CYUV
Sorenson Video 1	X	X	Fourcc: SVQ1
Sorenson Video 3		X	Fourcc: SVQ3
On2 VP3		X	Still experimental
Theora		X	Still experimental
Intel Indeo 3		X	Only works on i386 right now
FLV	X	X	Sorenson H. 263used in Flash
ATI VCR1		X	Fourcc: VCR1
ATI VCR2		X	Fourcc: VCR2
Cirrus Logic AccuPak		X	Fourcc: CLJR
4X video		X	Used in certain computer games
Sony Playstation MDEC		X	
Id RoQ		X	Used in Quake III, Jedi Knight II, other computer games
Xan/WC3		X	Used in Wing Commander III .MVE files
Interplay video		X	Used in Interplay .MVE files
Apple Animation		X	Fourcc: ‘rle’
Apple Graphics		X	Fourcc: ‘smc’
Apple Video		X	Fourcc: rpza
Apple Quickdraw		X	Fourcc: qdrw
Cinepak		X	
Microsoft RLE		X	
Microsoft Video-1		X	
Westwood VQA		X	
Id Cinematic Video		X	Used in Quake II
Planar RGB		X	Fourcc: 8BPS
FLIC Video		X	
Duck TrueMotion v1		X	Fourcc: DUCK
VMD Video		X	Used in Sierra VMD files
MSZH		X	Part of LCL
ZLIB	X	X	Part of LCL, encoder experimental
TechSmith Camtasia		X	Fourcc: TSCC
IBM Ultimotion		X	Fourcc: ULTI
Miro VideoXL		X	Fourcc: VIXL

23.2 FOBS

FOBS is an object-oriented wrapper for FFMPEG library (FOBS are Ffmeg OBjectS). It is a set of object oriented APIs to deal with media. It relies on the FFMPEG library but provides a much simpler programming interface. FOBS is currently available in C++ and has been successfully tested in a range of platforms (Linux, Max OS, Win32 (MinGW)). FOBS is released under the LGPL licence.

Here is a short C++ example of how to transcode a video file using FOBS’ API. The example transcodes an “avi” file into an “mp4”. The examples illustrates how to set the parameters of the transcoded output file (bit rate, frame rate, number of audio channels…).

```
#include <iostream>
#include “transcoder.h”
```

```
int main()
{
omnivedia::fobs::returnCode          error;
std::string                          inputFile( “test.avi” );
std::string                          outputFile( “test.mp4” );

// create transcoder object
omnivedia::fobs::transcoder          t ( inputFile.c_str(), outputFile.c_str() );

// choose output video codec:
//        - width:         352 pixels
//        - height:        288 pixels
//        - bit rate:      400 kb/s
//        - frame rate:    25 f/s
//        - codec:         msmpeg4
error = t.chooseVideoCodex( 352, 288, 400, 25, “msmpeg4” );
if( isError( error ) ) {
        std::cout << “Error choosing video codec” << std::endl;
        exit(-1);
}

// choose output audio codec:
//        - samples per second:        44100
//        - number of channels:        2
//        - bit rate:                  64 kb/s
//        - codec:                     mp2
error = t.chooseAudioCodec( 44100, 2, 64, “mp2” );
if( isError( error ) ) {
        std::cout << “Error choosing audio codec” << std::endl;
        exit(-1);
}

// choose output file format:
//        - output file format:        mp4
error = t.chooseFormat( “mp4” );
if( isError( error ) ) {
        std::cout << “Error choosing file format” << std::endl;
        exit(-1);
}

// perform the actual transcoding:
error = t.transcode();
if( isError( error ) ) {
        std::cout << “Error in transcoding” << std::endl;
        exit(-1);
}

return 0;

}
```

23.3 ImageMagick

ImageMagick™, version 6.1.9-4 (http://www.imagemagick.org), is a free software suite for the creation, modification and display of bitmap images. It can read, convert and write images in a large variety of formats. Images can be cropped, colors can be changed, various effects can be applied, images can be rotated and combined, and text, lines, polygons, ellipses and Bézier curves can be added to images and stretched and rotated. ImageMagick is free software: it is delivered with full source code and can be freely used, copied, modified and distributed. Its license is compatible with the GPL. ImageMagick is available for free, may be used to support both open and proprietary applications, and may be redistributed without fee. It runs on all major operating systems. Most of the functionality of ImageMagick can be used interactively from the command line; more often, however, the features are used from programs written in the programming languages Perl, C, C++, Python, PHP, Ruby or Java, for which ready-made ImageMagick interfaces (PerlMagick, Magick++, PythonMagick, MagickWand for PHP, RubyMagick, and JMagick) are available. This makes it possible to modify or create images automatically and dynamically.

Features and Capabilities - Here are just a few examples of what ImageMagick can do:

5. Convert an image from one format to another (e.g. TIFF to JPEG)
6. Resize, rotate, sharpen, color reduce, or ali special effects to an image
7. Create a montage of image thumbnails
8. Create a transparent image suitable for use on the Web
9. Turn a group of image into a GIF animation sequence
10. Create a composite image by combining several separate image
11. Draw shapes or text on an image
12. Decorate an image with a border or frame
13. Describe the format and characteristics of an image

ImageMagick includes a number of ready-made ImageMagick interfaces. This makes it possible to modify or create images automatically and dynamically. The following table shows supports to different programming languages.

Programming language	Tool/library
C	Use MagickWand to convert, compose, and edit images from the C language. There is also the low-level MagickCore library but is only recommended for wizard-level developers.
C++	Magick++ provides an object-oriented C++ interface to ImageMagick.
Java	JMagick provides an object-oriented Java interface to ImageMagick.
Perl	Use PerlMagick to convert, compose, and edit images from the Perl language.
PHP	MagickWand for PHP a native PHP-extension to the ImageMagick MagickWand API.
Python	PythonMagick an object-oriented Python interface to ImageMagick.
Ruby	RubyMagick is an interface between the Ruby programming language and the ImageMagick image processing libraries.

Supported Image Formats - ImageMagick supports reading over 90 major file formats (not including sub-formats). The following table provides a summary of the supported image formats. The *Mode* column reports the availability to read and/or write the format.

Tag	Mode	Description	Notes
ART	R	PFS: 1st Publisher	Format originally used on the Macintosh (MacPaint?) and later used for PFS: 1st Publisher clip art.
AVI	R	Microsoft Audio/Visual Interleaved	
AVS	RW	AVS X image	
BMP	RW	Microsoft Windows bitmap	

CGM	R	Computer Graphics Metafile	Requires ralcgm to render CGM files.
CIN	RW	Kodak Cineon Image Format	Cineon Image Format is a subset of SMTPE DPX.
CMYK	RW	Raw cyan, magenta, yellow, and black samples	Set `-size` and `-depth` to specify the image width, height, and depth.
CMYKA	RW	Raw cyan, magenta, yellow, black, and alpha samples	Set `-size` and `-depth` to specify the image width, height, and depth.
CUR	R	Microsoft Cursor Icon	
CUT	R	DR Halo	
DCM	R	Digital Imaging and Communications in Medicine (DICOM) image	Used by the medical community for images like X-rays.
DCX	RW	ZSoft IBM PC multi-page Paintbrush image	
DIB	RW	Microsoft Windows Device Independent Bitmap	DIB is a BMP file without the BMP header. Used to support embedded images in compound formats like WMF.
DPX	RW	Digital Moving Picture Exchange	
EMF	R	Microsoft Enhanced Metafile (32-bit)	Only available under Microsoft Windows.
EPDF	RW	Encapsulated Portable Document Format	
EPI	RW	Adobe Encapsulated PostScript Interchange format	Requires Ghostscript to read.
EPS	RW	Adobe Encapsulated PostScript	Requires Ghostscript to read.
EPS2	W	Adobe Level II Encapsulated PostScript	Requires Ghostscript to read.
EPS3	W	Adobe Level III Encapsulated PostScript	Requires Ghostscript to read.
EPSF	RW	Adobe Encapsulated PostScript	Requires Ghostscript to read.
EPSI	RW	Adobe Encapsulated PostScript Interchange format	Requires Ghostscript to read.
EPT	RW	Adobe Encapsulated PostScript Interchange format with TIFF preview	Requires Ghostscript to read.
FAX	RW	Group 3 TIFF	See TIFF format. Note that FAX machines use non-square pixels which are 1.5 times wider than they are tall but computer displays use square pixels so FAX images may appear to be narrow unless they are explicitly resized using a resize specification of "150x100%".
FIG	R	FIG graphics format	Requires TransFig.
FITS	RW	Flexible Image Transport System	
FPX	RW	FlashPix Format	Requires FlashPix SDK.
GIF	RW	CompuServe Graphics Interchange Format	8-bit RGB PseudoColor with up to 256 palette entires. Specify the format "GIF87" to write the older version 87a of the format.
GPLT	R	Gnuplot plot files	Requires gnuplot3.5.tar.Z or later.
GRAY	RW	Raw gray samples	Use `-size` and `-depth` to specify the image width, height, and depth.

HPGL	R	HP-GL plotter language	Requires hp2xx-3.2.0.tar.gz
HTML	RW	Hypertext Markup Language with a client-side image map	Also known as "HTM". Requires html2ps to read.
ICO	R	Microsoft icon	Also known as "ICON".
JBIG	RW	Joint Bi-level Image experts Group file interchange format	Also known as "BIE" and "JBG". Requires jbigkit-1.0.tar.gz.
JNG	RW	Multiple-image Network Graphics	JPEG in a PNG-style wrapper with transparency. Requires libjpeg and libpng-1.0.2 or later, libpng-1.2.5 or later recommended.
JP2	RW	JPEG-2000 JP2 File Format Syntax	Requires jasper-1.600.0.zip
JPC	RW	JPEG-2000 Code Stream Syntax	Requires jasper-1.600.0.zip
JPEG	RW	Joint Photographic Experts Group JFIF format	Requires jpegsrc.v6b.tar.gz
MAN	R	Unix reference manual pages	Requires that GNU groff and Ghostcript are installed.
MAT	R	MATLAB image format	
MIFF	RW	Magick image file format	Open ImageMagick's own image format (with ASCII header) which ensures that no image attributes understood by ImageMagick are lost.
MONO	RW	Bi-level bitmap in least-significant-byte first order	
MNG	RW	Multiple-image Network Graphics	A PNG-like Image Format Supporting Multiple Images, Animation and Transparent JPEG. Requires libpng-1.0.2 or later, libpng-1.2.5 or later recommended.
MPEG	RW	Motion Picture Experts Group file interchange format (version 1)	Requires mpeg2vidcodec_v12.tar.gz.
M2V	RW	Motion Picture Experts Group file interchange format (version 2)	Requires mpeg2vidcodec_v12.tar.gz.
MPC	RW	Magick Persistent Cache image file format	The native "in-memory" ImageMagick uncompressed file format. This file format is identical to that used by Open ImageMagick to represent images in memory and is read in "zero time" via memory mapping. The MPC format is not portable and is not suitable as an archive format. It is suitable as an intermediate format for high-performance image processing. The MPC format requires two files to support one image. When writing the MPC format, a file with extension ".mpc" is used to store information about the image, while a file with extension “.cache” stores the image pixels. The storage space required by a MPC image (or an image in memory) may be calculated by the equation (5*QuantumDepth*Rows*Columns)/8.
MSL	RW	Magick Scripting Language	MSL is the XML-based scripting language supported by the conjure utility.
MTV	RW	MTV Raytracing image format	
MVG	RW	Magick Vector Graphics.	The native ImageMagick vector metafile format. A text file containing vector drawing commands accepted by convert's -draw option.
OTB	RW	On-the-air Bitmap	
P7	RW	Xv's Visual Schnauzer thumbnail format	

PALM	RW	Palm pixmap	
PBM	RW	Portable bitmap format (black and white)	
PCD	RW	Photo CD	The maximum resolution written is 768x512 pixels since larger images require huffman compression (which is not supported).
PCDS	RW	Photo CD	Decode with the sRGB color tables.
PCL	W	HP Page Control Language	For output to HP laser printers.
PCX	RW	ZSoft IBM PC Paintbrush file	
PDB	RW	Palm Database ImageViewer Format	
PDF	RW	Portable Document Format	Requires Ghostscript to read.
PFA	R	Postscript Type 1 font (ASCII)	Opening as file returns a preview image.
PFB	R	Postscript Type 1 font (binary)	Opening as file returns a preview image.
PGM	RW	Portable graymap format (gray scale)	
PICON	RW	Personal Icon	
PICT	RW	Apple Macintosh QuickDraw/PICT file	
PIX	R	Alias/Wavefront RLE image format	
PNG	RW	Portable Network Graphics	Requires libpng-1.0.2 or later, libpng-1.2.5 or later recommended.
PNM	RW	Portable anymap	PNM is a family of formats supporting portable bitmaps (PBM) , graymaps (PGM), and pixmaps (PPM). There is no file format associated with pnm itself. If PNM is used as the output format specifier, then ImageMagick automatically selects the most appropriate format to represent the image. The default is to write the binary version of the formats. Use `+compress` to write the ASCII version of the formats.
PPM	RW	Portable pixmap format (color)	
PS	RW	Adobe PostScript file	Requires Ghostscript to read.
PS2	RW	Adobe Level II PostScript file	Requires Ghostscript to read.
PS3	RW	Adobe Level III PostScript file	Requires Ghostscript to read.
PSD	RW	Adobe Photoshop bitmap file	
PTIF	RW	Pyramid encoded TIFF	Multi-resolution TIFF containing successively smaller versions of the image down to the size of an icon. The desired sub-image size may be specified when reading via the `-size` option.
PWP	R	Seattle File Works multi-image file	
RAD	R	Radiance image file	Requires that *ra_ppm* from the Radiance software package be installed.
RGB	RW	Raw red, green, and blue samples	Use `-size` and `-depth` to specify the image width, height, and depth.
RGBA	RW	Raw red, green, blue, and	Use `-size` and `-depth` to specify the image width,

		alpha samples	height, and depth.
RLA	R	Alias/Wavefront image file	
RLE	R	Utah Run length encoded image file	
SCT	R	Scitex Continuous Tone Picture	
SFW	R	Seattle File Works image	
SGI	RW	Irix RGB image	
SHTML	W	Hypertext Markup Language client-side image map	Used to write HTML clickable image maps based on a the output of montage or a format which supports tiled images such as MIFF.
SUN	RW	SUN Rasterfile	
SVG	RW	Scalable Vector Graphics	Requires libxml2 and freetype-2. Note that SVG is a very complex specification so support is still not complete.
TGA	RW	Truevision Targa image	Also known as formats "ICB", "VDA", and "VST".
TIFF	RW	Tagged Image File Format	Also known as "TIF". Requires tiff-v3.6.1.tar.gz or later. Note that since Unisys claims a patent on the LZW algorithm (expiring in the US as of June 2003) used by LZW-compressed TIFF files, ImageMagick binary distributions do not include support for the LZW algorithm so LZW TIFF files can not be written. Although a patch is available for libtiff to enable building with LZW support. Users should consult the Unisys LZW web page before applying it.
TIM	R	PSX TIM file	
TTF	R	TrueType font file	Requires freetype 2. Opening as file returns a preview image.
TXT	RW	Raw text file	
UIL	W	X-Motif UIL table	
UYVY	RW	Interleaved YUV raw image	Use `-size` command line option to specify width and height.
VICAR	RW	VICAR rasterfile format	
VIFF	RW	Khoros Visualization Image File Format	
WBMP	RW	Wireless bitmap	Support for uncompressed monochrome only.
WMF	R	Windows Metafile	Requires libwmf. By default, renders WMF files using the dimensions specified by the metafile header. Use the -density option to adjust the output resolution, and thereby adjust the ouput size. The default output resolution is 72DPI so "-density 144" results in an image twice as large as the default. Use **-background color** to specify the WMF background color (default white) or **-texture filename** to specify a background texture image.
WPG	R	Word Perfect Graphics File	
XBM	RW	X Windows system bitmap, black and white only	Used by the X Windows System to store monochrome icons.
XCF	R	GIMP image	
XPM	RW	X Windows system pixmap	Also known as "PM". Used by the X Windows System to store color icons.
XWD	RW	X Windows system window dump	Used by the X Windows System to save/display screen dumps.
YCbCr	RW	Raw Y, Cb, and Cr samples	Use `-size` and `-depth` to specify the image width, height, and depth.
YCbCrA	RW	Raw Y, Cb, Cr, and alpha samples	Use `-size` and `-depth` to specify the image width,

			height, and depth.
YUV	RW	CCIR 601 4:1:1	

23.4 LIBSNDFILE

Libsndfile is a C library for reading and writing files containing sampled sound (such as MS Windows WAV and the Apple/SGI AIFF format) through one standard library interface. It is released in source code format under the Gnu Lesser General Public License.

The library was written to compile and run on a Linux system but should compile and run on just about any Unix (including MacOSX). It can also be compiled and run on Win32 systems using the Microsoft compiler and MacOS (OS9 and earlier) using the Metrowerks compiler. There are directions for compiling libsndfile on these platforms in the Win32 and MacOS directories of the source code distribution.

It was designed to handle both little-endian (such as WAV) and big-endian (such as AIFF) data, and to compile and run correctly on little-endian (such as Intel and DEC/Compaq Alpha) processor systems as well as big-endian processor systems such as Motorola 68k, Power PC, MIPS and Sparc. Hopefully the design of the library will also make it easy to extend for reading and writing new sound file formats.

It has been compiled and tested (at one time or another) on the following systems:

- i586-pc-linux-gnu (Linux on PC hardware)
- powerpc-unknown-linux-gnu (Linux on Apple Mac hardware)
- powerpc-apple-darwin7.0 (Mac OS X 10.3)
- sparc-sun-solaris2.8 (using gcc)
- mips-sgi-irix5.3 (using gcc)
- QNX 6.0
- i386-unknown-openbsd2.9
- Win32 (Microsoft Visual C++)

At the moment, each new release is being tested on i386 Linux, PowerPC Linux, MacOSX on PowerPC and Win32.
Features
libsndfile has the following main features :

Ability to read and write a large number of file formats.

- A simple, elegant and easy to use Applications Programming Interface.
- Usable on Unix, Win32, MacOS and others.
- On the fly format conversion, including endian-ness swapping, type conversion and bitwidth scaling.
- Optional normalisation when reading floating point data from files containing integer data.
- Ability to open files in read/write mode.
- The ability to write the file header without closing the file (only on files open for write or read/write).
- Ability to query the library about all supported formats and retrieve text strings describing each format.

libsndfile has a comprehensive test suite so that each release is as bug free as possible. When new bugs are found, new tests are added to the test suite to ensure that these bugs don't creep back into the code. When new features are added, tests are added to the test suite to make sure that these features continue to work correctly even when they are old features.

The following table lists the file formats and encodings that libsndfile can read and write. The file formats are arranged across the top and encodings along the left edge.

	Micro-soft WAV	SGI / Apple AIFF / AIFC	Sun / DEC / NeXT AU / SND	Header-less RAW	Paris Audio File PAF	Commo-dore Amiga IFF / SVX	Sphere Nist WAV	IRCAM SF	Creative VOC	Sound forge W64	GNU Octave 2.0 MAT4	GNU Octave 2.1 MAT5	Portable Voice Format PVF	Fasttracker 2 XI	HMM Tool Kit HTK
Unsigned 8 bit PCM	R/W	R/W		R/W					R/W	R/W		R/W			
Signed 8 bit PCM		R/W	R/W	R/W	R/W	R/W	R/W						R/W		
Signed 16 bit PCM	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Signed 24 bit PCM	R/W	R/W	R/W	R/W	R/W		R/W	R/W		R/W					
Signed 32 bit PCM	R/W	R/W	R/W	R/W			R/W	R/W		R/W	R/W	R/W	R/W		
32 bit float	R/W	R/W	R/W	R/W				R/W		R/W	R/W	R/W			
64 bit double	R/W	R/W	R/W	R/W						R/W	R/W	R/W			
u-law encoding	R/W	R/W	R/W	R/W			R/W	R/W	R/W	R/W					
A-law encoding	R/W	R/W	R/W	R/W			R/W	R/W	R/W	R/W					
IMA ADPCM	R/W									R/W					
MS ADPCM	R/W									R/W					
GSM 6.10	R/W	R/W		R/W						R/W					
G721 ADPCM 32kbps			R/W												
G723 ADPCM 24kbps			R/W												

G723 ADPCM 40kbps			R/W												
12 bit DWVW		R/W		R/W											
16 bit DWVW		R/W		R/W											
24 bit DWVW		R/W		R/W											
Ok Dialogic ADPCM				R/W											
8 bit DPCM														R/W	
16 bit DPCM														R/W	

24 MPEG-7: eXperimental Model (XM)

This subsection describes the complete video description functionality that is available in the MPEG-7 eXperimental Model (XM). The software (Unix and windows version) and the required libraries can be download at http://www.lis.ei.tum.de/research/bv/topics/mmdb/mpeg7.html

24.1.1 3DShapeSpectrum

Technique	3D Shape Spectrum Descriptor
Document	Text of ISO/IEC CD 15938-3 Multimedia Content Description Interface - Part 3 Visual, Chapter "The 3D Shape Spectrum Descriptor"
Name	Titus Zaharia, Institut National des Telecommunications
EMail	titus.zaharia@int-evry.fr
Type	Application
External Libraries	NONE
Related Ds/DSs	2D Shape Descriptors, Multiview Descriptor
Used Ds/DSs	NONE
Input	3D Meshes in VRML2.0 format
Extraction	Yes
Client Appl	Search & Retrieval
Summary	The 3D Shape Spectrum Descriptor provides a compact shape description of 3D data represented as 3D meshes. The 3D Shape Spectrum Descriptor is the histogram (distribution) of the shape index over the entire mesh. The shape index is defined as the angular coordinate of a polar representation of the principal curvature vector. The descriptor includes two additional components quantifying the amount of singular components (e.g. borders) and planar components (for which the shape index is not defined).
Strong Points	Compactness
Limitations	None
Known Problems	None
Parameters	NoOfBins : represents the number of bins used for the histogram representation. NoOfBins may take any positive integer value. Usual values range in the [10, 100] interval. NoOfBits : represents the number of bits that uniformly quantize the 3D Shape Spectrum values. Usual values: 8, 9, 10, 11, 12, 13, 14. Metric : specifies the metric used for similarity retrieval. The possible values and their associated metrics are listed here below: 0 - L1-based distance, without the singular and planar surface components 1 - L2-based distance, without the singular and planar surface components 2 - pseudo-weighted similarity measure, without the singular and planar surface components 3 - L1-based distance, without the singular surface components 4 - L2-based distance, without the singular surface components 5 - pseudo-weighted similarity measure, without the singularr surface components 6 - L1-based distance, with all components 7 - L2-based distance, with all the components 8 - pseudo-weighted similarity measure, with all components Example: NoOfBins 100 NoOfBits 12

Metric 0

24.1.2 AdvancedFaceRecognition

Technique	AdvancedFaceRecognition
Document	MPEG-7/Visual part of eXperimental Model Version17, see AdvancedFaceRecognition Descriptor
Name	Toshio Kamei, NEC Corporation
EMail	t-kamei@cb.jp.nec.com
Contact	t-kamei@cb.jp.nec.com
Type	Application
External Libraries	none
Related Ds/DSs	none
Used Ds/DSs	none
Input	Images
Extraction	Yes
Client Appl	Search & Retrieval
Summary	AdvancedFaceRecognition is a descriptor of face identity robust to variations in pose and illumination conditions.
Strong Points	compact representation of facail identity, and high-speed and accurate matching.
Limitations	The face images should be normalized before feature extraction. The positions of two eyes should be at (24,16) and (24,31) in the scaled image(56 pixels in height and 46 pixels in width)
Known Problems	none
Parameters	This descriptor supports scalable represenation of facial feature vector. If you wish to change the dimensionality of the vector, set extraction parameters and matching parameters. The allowed range of the extraction parameters is from 24 to 63 for FourierFeature, and from 0 to 63 for CompositeFeature. The dimensions of the feature vectors in matching must not exceed those of the extracted feature vectors.

24.1.3 CameraMotion

Technique	CameraMotionType
Document	W3703 " CD 15938-3 MPEG-7 Multimedia Content Description Interface - Part 3 Visual", La Baule meeting, describes the normative parts of CameraMotion. W3673 "MPEG-7 visual part of XM 8.0", La Baule meeting, describes the non-normative parts of CameraMotion
Name	Benoit Mory, Philips Recherche France
EMail	benoit.mory@philips.com
Type	Descriptor
External Libraries	N/A
Related Ds/DSs	MotionActivity, MotionTrajectory
Used Ds/DSs	None
Input	Mpeg video stream segments (e.g. foo.mpg[1-1000])
Extraction	Yes
Client Appl	Search & Retrieval
Summary	This descriptor characterizes qualitatively and quantitatively camera motion parameters along time (typically along a video segment). It is based on 3-D camera motion parameter information, which can be automatically extracted or generated by capture devices.The descriptor supports any combination of the following well-known basic camera operations : fixed, panning, tracking, tilting, booming, zooming, dollying, and rolling.
Strong Points	As a discriminant feature (motion) of video streams, the camera motion is typically useful in low-level feature-based retrieval, combined with spatial descriptors such as colour, texture or shape. Could also be used in professional applications.

Limitations	The extraction method implemented in the XM does not allow to discriminate between pans and tracks (rotations and translations) when the visual field is small (same motion vector fields).
Known Problems	None
Parameters	None?

24.1.4 ColorLayout

Technique	Color Layout
Document	MPEG-7 Visual XM and CD, see Color Layout Descriptor
Name	Akio YAMADA (NEC Corp.)
EMail	a-yamada@da.jp.nec.com
Contact	a-yamada@da.jp.nec.com
Type	Application
External Libraries	none
Related Ds/DSs	Still Region DS, Video Segment DS, Moving Object Ds, TimeSeries DS
Used Ds/DSs	none
Input	Images
Extraction	Yes
Client Appl	Search & Retrieval
Summary	Color Layout is designed to achieve very high-speed retrieval of images or video frames based on the spatial distribution of color. This library supports both extraction (rectangular and arbitrary shaped) and matching (see ClientApplication as an example). To implement the video segment retrieval functionality, this descriptor should be contained in TimeSeries DS (see TimeSeries DS).
Strong Points	High-speed image and video segment retrieval.Very compact representation of annotation data.(request only 8 Bytes for each image)
Limitations	None. The components had been tested using over 50,000 digital photos.
Known Problems	none
Parameters	Color Layout descriptor supports scalable representation of spatial distribution of color. If you wish to change the number of coefficients, you should directly modify the parameters of StartExtraction()

24.1.5 ColorQuantization

Technique	Color Quantization
Document	MPEG-7 Visual XM and CD, see Color Quantization Descriptor
Name	Jungmin Song, Heon Jun Kim
EMail	jmsong73@mail.lgcit.com, hjk@lge.co.kr
Contact	Jungmin Song
Type	descriptor for dominant color descriptor
External Libraries	none
Related Ds/DSs	Dominant Color Descriptor and DS related to Dominant Color D.
Used Ds/DSs	none
Input	n/a
Extraction	n/a
Client Appl	n/a
Summary	This descriptor defines the uniform quantization of a color space when the color space is quantized uniformly in all color components. (color space is defined by Color Space Descriptor)
Limitations	Non uniform color quantization is not currently supported.
Known Problems	None.
Parameters	quantization components and bin numbers

24.1.6 ColorSpace

Technique	Color Space
Document	MPEG-7 Visual XM and CD, see Color Space Descriptor
Name	Jungmin Song, Heon Jun Kim
EMail	jmsong73@mail.lgcit.com, hjk@lge.co.kr
Contact	Jungmin Song
Type	descriptor for dominant color descriptor
External Libraries	none
Related Ds/DSs	Dominant Color Descriptor and DS related to Dominant Color D.
Used Ds/DSs	none
Input	n/a
Extraction	n/a but some color transform modules between different color space are provided in ColorSpaceExtraction.cpp
Client Appl	n/a
Summary	This descriptor specifies the color space that is to be used in other color based descriptions. In the current description, the following color spaces are supported: - RGB - YCbCr - HSV - HMMD - Linear transformation matrix with reference to RGB - Monochrome Color transform modules provided in extraction (ColorSpaceExtraction.cpp) are as follows: -YUV(YCbCr) to HSV -YUV(YCbCr) to RGB -RGB to YUV(YCbCr) -RGB to HSV -HSV to RGB -RGB to HMMD -HMMD to RGB
Limitations	Currently, the descriptor specifies only limited number of color spaces which are RGB, YCbCr, HSV, HMMD, Linear transformation matrix with reference to RGB and Monochrome.
Known Problems	None.
Parameters	color space index

24.1.7 ColorStructure

Technique	ColorStructure
Document	W3703, Sect 6.6
Name	Jim Errico, Sharp Labs of America; Dean Messing, Sharp Labs of America
EMail	jerrico@sharplabs.com; deanm@sharplabs.com
Type	Application
External Libraries	NA
Related Ds/DSs	NA
Used Ds/DSs	NA
Input	Images
Extraction	Yes
Client Appl	Search & Retrieval
Summary	This component is the implementation of the extraction and search functionality for the ColorStructure Descriptor.
Strong Points	NA
Limitations	NA
Known Problems	NA
Parameters	ColorQuantSize : one of {256, 128, 64, 32}

24.1.8 ContourShape

Technique	Contour shape descriptor
Document	MPEG-7 Committee Draft Visual, see Contour shape descriptor
Name	Miroslaw Bober (Mitsubishi Electric ITE)
EMail	miroslaw.bober@vil.ite.mee.com
Contact	miroslaw.bober@vil.ite.mee.com
Type	Application
External Libraries	none
Related Ds/DSs	Still region DS
Used Ds/DSs	none
Input	Binary shape masks
Extraction	Yes
Client Appl	Search & Retrieval
Summary	The Contour-based shape descriptor describes shape features of a closed contour. It uses the curvature scale space representation and eccentricity and circularity of the original and filtered shape.
Strong Points	The descriptor is rotational, scale and position invariant. It is robust to affine shape deformations and deformations due to non-rigid motion (both very common in video sequences) The descriptor has good shape generalisation properties, similar to human perception of visual shapes. It is very compact (110 bits on average for the MPEG-7 shape database).
Limitations	Here are some limitations of the current implementation: Default white shape on black background (can be changed) If more than one contour is in the image, the descriptor is extracted from the first contour found in the image (upper-left, top left).
Known Problems	None
Parameters	Uses the general XM-SW parameters.

24.1.9 DominantColor

Technique	Dominant Color
Document	MPEG-7 Visual XM and CD, see Dominant Color Descriptor
Name	Jungmin Song, Heon Jun Kim, Leszek Cieplinski, Prof. Manjunath
	EMail jmsong73@mail.lgcit.com, hjk@lge.co.kr, Leszek.Cieplinski@vil.ite.mee.com, manj@ece.ucsb.edu
Contact	Jungmin Song(general), Leszek Cieplinski(color variance), Prof. Manjunath(dominant color extraction and search algorithm)
Type	application
External Libraries	none
Related Ds/DSs	Color Space Descriptor, Color Quantization Descriptor, Related DS's are not defined fully yet.
Used Ds/DSs	Color Space Descriptor, Color Quantization Descriptor
Input	Images
Extraction	Yes
Client Appl	Search, Retrieval, Browsing
Summary	The dominant color descriptor is useful for image and video retrieval. It targets content-based retrieval for color, either for the whole image or for any arbitrary shaped region (rectangular or irregular). It is a very compact descriptor, requiring less than 6-8 colors per region. Since colors are not pre-quantized as in the histogram type color descriptors, the representation is more accurate. To accomplish high accuracy in retrieval, Spatial Coherency and/or Color Variance can be utilized. It is intended for applications that use object based representations (objects or regions in an image).
Limitations	The maximum allowed number of dominant colors is 8.
Known Problems	None.

Parameters	color space and color quantization parameters

24.1.10 EdgeHistogram

Technique	Edge Histogram
Document	MPEG-7 Visual XM and CD, see Edge Histogram Descriptor
Name	Soo-Jun Park (ETRI), Chee Sun Won (Dongguk Uni.)
EMail	psj@etri.re.kr, cswon@dongguk.edu
Contact	psj@etri.re.kr
Type	Application
External Libraries	none
Related Ds/DSs	Still Region DS, Video Segment DS, Moving Object Ds, TimeSeries DS
Used Ds/DSs	none
Input	Images
Extraction	Yes
Client Appl	Search & Retrieval
Summary	This library implements the edge histogram descriptor, which is designed to represent the spatial distribution of 5 types of edges in local image regions. It includes both extraction (rectangular and arbitrary shaped) and matching. For the non-normative matching, it includes the semi-global and global histogram matching schemes. For the video segment retrieval, this descriptor should be contained in TimeSeries DS.
Strong Points	Since the edge histogram descriptor consists of local edge histograms only, it is very flexible to represent some global and semi-global edge characteristic for the matching and is also compact.
Limitations	None Although our recommendation for the smallest image/object size is 70*70(pixels), this library also works for the images/objects smaller than that by exapnding the image/object size with a bilinear interpoaltion scheme.
Known Problems	None.
Parameters	None.

24.1.11 FaceRecognition

Technique	FaceRecognition
Document	Text of ISO/IEC CD 15938-3 Multimedia Content Description Interface - Part 3 Visual
Name	Lei Wang, Panasonic Singapore Labs; Mark Pickering, Univ. of New South Wales
EMail	Lwang@psl.com.sg, m-pickering@adfa.edu.au
Contact	Lwang@psl.com.sg, m-pickering@adfa.edu.au
Type	Application
External Libraries	NONE
Related Ds/DSs	NONE
Used Ds/DSs	NONE
Input	Images(JPEG,BMP,etc)
Extraction	Yes
Client Appl	Search & Retrieval
Summary	This component generates a FaceRecognition descriptor for face recognition and query. There is no other function related to this component. The descriptor uses eigenface method to extract face features.
Strong Points	Computational simplicity and accuracy
Limitations	The face images should be normalized before feature extraction. The positions of two eyes should be at (24,16) and (24,31) in the scaled image(56 pixels in height and 46 pixels in width)
Known Problems	NONE
Parameters	NONE

24.1.12 GoFGoPColor

Technique	GoF/GoP Color Descriptor
Document	MPEG-7 Visual XM and CD, see GoF/GoP Color
Name	Santhana Krishnamachari, Mufit Ferman
Contact	santhana.krishnamachari@philips.com
Type	Application
External Libraries	None
Related Ds/DSs	Scalable Color, Video Segment DS, Collection DS
Used Ds/DSs	Scalable Color
Input	Video, Image
Extraction	Yes
Client Appl	Search & Retrieval
Summary	This library implements the GoF/GoP Color descriptor which is used to describe the color characteristics of a collection of video frames (and a collection of images). consists of one primary and four secondary attributes. Since the feature vector is short, a simple absolute distance or squared distance criterion can be used for matching.
Strong Points	None.
Limitations	Use mean or median aggregation for matching
Known Problems	None.
Parameters	None.

24.1.13 GridLayout

Technique	Grid Layout
Document	ISO/IEC 15938-3/-8, see Grid Layout Descriptor
Name	Akio YAMADA (NEC Corp.)
EMail	a-yamada@da.jp.nec.com
Contact	a-yamada@da.jp.nec.com
Type	Basic Component
External Libraries	none
Related Ds/DSs	Still Region DS
Used Ds/DSs	none
Input	Images
Extraction	no
Client Appl	n/a
Summary	Grid layout is one of the container datatype carrying several visual descriptors. It splits an image into several grid and we can assign the visual Ds on each geid.
Strong Points	n/a
Limitations	depends on the contained descriptor
Known Problems	none
Parameters	depends on the contained descriptor

24.1.14 HomoTexture

Technique	HomogeneousTexture
Document	Text of ISO/IEC CD 15938-3 Multimedia Content Description Interface - Part 3 Visual(w4062), Chapter 7.1
Name	yanglim Choi, M/M Lab, Samsung Advanced Institute of Technology
EMail	yanglimc@samsung.com
Type	Application
External Libraries	NONE
Related Ds/DSs	StillRegionDS, StillImageDS, MovingRegionDS
Used Ds/DSs	NONE
Input	Images/Regions(JPEG,BMP,etc)
Extraction	Yes

Client Appl	Search & Retrieval
Summary	This component generates a texture descriptor for a homogeneously textured Image/Region for Search and Retrieval using texture features. The descriptor components are the average and the standard deviation of the image together with the energies and the energy deviations of Gabor filtered reponse of each (predefined) fequency channels.
Strong Points	Statistically very precise description of homogeneously textured region.
Limitations	Relatively large in size (32 components for the basic layer and 62 components for the enhanced layer).
Known Problems	NONE
Parameters	HomogeneousTexture.par: dummy parameter file to show the usage /* additional parameters*/ layer : The layer information of the descriptor 0:base-layer, 32 components 1:full-layer, 62 components option: Matching option for client n: normal matching r:rotation invariant matching s:scale invariant matching rs:rotation and scale invariant matching

24.1.15 MotionActivity

Technique	Motion Activity
Document	MPEG-7 Visual XM and CD, see Motion Activity Descriptor
Name	Ajay Divakaran (Mitsubishi Electric Research Labs.)
EMail	ajayd@merl.com
Contact	ajayd@merl.com
Type	Application
External Libraries	none
Related Ds/DSs	Video Segment DS
Used Ds/DSs	none
Input	Video
Extraction	Yes
Client Appl	Search & Retrieval
Summary	This library implements the motion activity descriptor which consists of one primary and four secondary attributes. The primary attribute is the intensity of motion activity, and the other four attributes express the dominant direction, the spatial distribution, the temporal distribution and the spatial localization of motion activity. Since the feature vector is short, a simple absolute distance or squared distance criterion can be used for matching.
Strong Points	The descriptor is compact and extremely effective as a filter for reduction of large search spaces.
Limitations	The program depends on previously computed motion vectors. It does not have its own motion estimation. This was done to maintain format independence.
Known Problems	None.
Parameters	None.

24.1.16 MotionTrajectory

Technique	MotionTrajectory
Document	W3703 " CD 15938-3 MPEG-7 Multimedia Content Description Interface - Part 3 Visual", La Baule meeting, describes the normative parts of MotionTrajectory. W3673 "MPEG-7 visual part of XM 8.0", La Baule meeting, describes the non-normative parts of MotionTrajectory.

	M6597 "Motion Trajectory: Software Modules and Content Set Information", S. Jeannin, La Baule meeting. The document complements and documents the contributions that have been made concerning the MotionTrajectory software modules and the MotionTrajectory Content Set. It describes the type of content given, its format, and its organization. It documents the software modules for the client and server applications, which are non-normative and not fully described in XM and CD. It tries to give some answers to questions that new users of the software may raise. Many other documents can be found that contain details on MotionTrajectory. We don't list them here. Interested readers can look in Lancaster proposals (MotionTrajectory proposals from Philips and Tektronix), Seoul document about the merging of the proposals, from Seoul to Maui different documents on MotionTrajectory Core Experiments, and documents on restructuration of the Descriptor to adapt it to the evolution of the global MPEG-7 structure.
Name	Sylvie Jeannin, Philips Research USA
EMail	Sylvie.Jeannin@philips.com
Contact	Sylvie.Jeannin@philips.com
Type	Application
External Libraries	None
Related Ds/DSs	Spatio-Temporal Locator, TemporalInterpolation, Spatial2Dcoordinates, Time.
Used Ds/DSs	TemporalInterpolation is instantiated once each time MotionTrajectory is instantiated. Spatial2Dcoordinates is optionally included by reference in MotionTrajectory; the inclusion is mentioned by a one bit flag in MotionTrajectory. TimeDS is used indirectly as some of its components are instantiated within TemporalInterpolation, which is instantiated within MotionTrajectory.
Input	Text files containing formatted Key-point List
Input Format	Each input file should contain the list of coordinates (time and space) of one representative point of the object whose trajectory should be described. There should be one file per object, i.e. per trajectory to describe.(Recommended naming rule: files can be named by the name of the sequence the object comes from, the object number, and their extension be ".data".) ex: Let us assume "yard000.data" contains the successive spatio-temporal positions of the representative point of the object #0 in the sequence Yard. On each line of data files, there should be: time (milliseconds), x_position, y_position, z_position. The x and y spatial positions should be normalized by the corresponding image dimensions. Their origin is on the top left corner of the image. Value "-1.0" should be used when the position is not known. ex: "2520 0.980114 0.541667 -1.000000" means that at t=2520ms, the representative point was at position (x=0.980114, y=0.541667) where x and y are normalized by the image size, and that its position in the third spatial dimension, z, is not known.Examples are provided in the MPEG-7 Content repository web site.
Extraction	Yes.
Client Appl	Search & retrieval: 12 different types of queries are implemented. The query type is specified by a parameter in the Client Application. They are intended to be examples, that demonstrate the expressiveness of the description and show how various query types can be added (depending on the application needs). All details about this can be found in m6597 (contribution to La Baule meeting). To summarize: the query types that are implemented in XM can be divided in two main categories: Similarity-based queries (low-level retrieval): For these queries, the program first extract the trajectory description from a given query data. It compares it to each description decoded from a given bitstream provided by the Server Application. Then it ranks the elements of the database whose trajectory is described in the bitstream from the most to the less similar to the query, in terms of their trajectory, and according to a specified similarity criteria.

	8 different similarity criteria can be used, that highlight different aspects of trajectory. Basically, we retained that trajectories can be similar in terms of positions, or/and speeds, or/and accelerations, and that time can also be handled differently in the criterion. Higher level queries: For these queries, no similarity is calculated: no input query data is needed; instead a high level criteria of the trajectory is given. The given bitstream provided by the Server Application is decoded, and the elements of the database whose trajectories are described in the bitstream are ranked, from the ones that fulfill the criteria the most to the ones that do not fulfill it at all. The examples of high-level query types that are implemented are: find objects moving to the right, to the left, find objects moving up, moving down.
Summary	Motion Trajectory is a high-level feature associated with a moving region, defined as a spatio-temporal localization of one of its representative points (such as the centroid). The software modules, integrated in the XM software, implement all parts of the MotionTrajectory descriptor, as what is described in Visual XM and CD. Corresponding C++ modules are parts of the "Descriptors", "ExtractionUtilities", "SearchUtilities", "CodingSchemes" and "Applications" (for both Server and Client) XM directories. It has been satisfactorily compiled and tested by independent parties on SUN/SOLARIS, PC/Windows and PC/LINUX.
Strong Points	Many different types of queries are implemented, both high-level and low-level. The program has been satisfactorily compiled and tested by independent parties on SUN/SOLARIS, PC/Windows and PC/LINUX.
Limitations	Warning for use: The input format for MotionTrajectory is a file containing successive spatio-temporal positions of a point, representative from the object whose trajectory needs to be described (see above). This was used in the Trajectory Core Experiments, and accepted in the Dec. 99 meeting as the input format for the XM software. This decision allows us to be independent from the way the object was detected, i.e. to remain generic in terms of the pre-extraction (segmentation) process. Indeed, different ways of detecting/specifying the object lead to different formats, and the first common format prior to trajectory characterization is the list of spatio-temporal positions. These positions though have to be generated off-line, in a process which is independent from XM. When a segmentation mask is available for defining the object, it is recommended to obtain the input file by simply calculating for each frame the center of mass of the segmentation mask, and storing these coordinates in a file using the format described above. This is an extremely straightforward process to implement. Of course other methods can be used to obtain successive positions of a representative point of an object whose trajectory should be described: positions can be directly obtained by segmentation of motion field, or by getting the centroid of a bounding box provided as output of a tracking program, etc.
Known Problems	Full integration with the Spatial2Dcoordinates, allowing their reference, and corresponding updates following La Baule is not yet provided.
Parameters	how to select the application??

24.1.17 MultiView

Technique	Multiple View
Document	nxxxx,MPEG-7 Committee Draft Visual, see Multiple View Descriptor
Name	Karsten Müller (HHI), James Cooper (Mitsubishi)
EMail	Kmueller@hhi.de, James.Cooper@vil.ite.mee.com

Contact	Kmueller@hhi.de
Type	Application
External Libraries	none
Related Ds/DSs	ContourShapeD, RegionShapeD, StillRegionDS (rectangular and arbitrarily shaped)
Used Ds/DSs	MultipleView D, ContourShapeD, RegionShapeD
Input	Number of Images (up to 16), originating from 3D object
Extraction	Yes
Client Appl	Search & Retrieval
Summary	Multiple View (as a container descriptor) uses the underlying Descriptor, e.g. ContourShape for Extraction and Retrieval of similar objects. Two retrieval methods are available: 3D-3D Retrieval, where a number of views from one object is compared to a number of views from the database objects and 2D-3D Retrieval, where a 2D-image is compared to the most similar views of the 3D-object
Strong Points	Depends on the underlying Descriptor
Limitations	Only up to 16 views per object are supported
Known Problems	None
Parameters	Uses the general XM-SW parameters, IMPORTANT: The list file should only contain the name of the original 3D object without extension, e.g. "c:\temp\model-1". The extensions for the views are appended automatically by the program. The convention is to name the 2D-views from that model "model-1_1.gif", "model-1_2.gif" up to "model-1_16.gif"

24.1.18 ParametricObjectMotion

Technique	Parametric Motion
Document	MPEG-7 Committee Draft Visual, see Parametric Motion Descriptor
Name	Aljoscha Smolic (HHI), Titus Zaharia (INT)
EMail	smolic@hhi.de
Contact	smolic@hhi.de
Type	Application
External Libraries	none
Related Ds/DSs	Video Segment DS, Moving Region DS, Mosaic DS, TimeSeries DS, Spatial 2D Coordinates D
Used Ds/DSs	none
Input	Pairs of images, the motion between each image pair is represented
Extraction	Yes
Client Appl	Search & Retrieval
Summary	Parametric Motion describes the global or object motion in video over a certain period of time, by a set of parameters according to a certain motion models. Different motion models, spatial and temporal references can be used, enabling high flexibility of possible descriptions. In addition to query-by-example (i.e. retrieving motions that are similar to a given example) it also enables query-by-specification of certain kinds of motion, like "find left translation" or "find up-scales (zoom)".
Strong Points	none
Limitations	none
Known Problems	none
Parameters	Uses the general XM-SW parameters. In the listfile image pairs must be specified that define the motion in between: motion_1_image_1 motion_1_image_2 motion_2_image_1 motion_2_image_2

:
motion_N_image_1
motion_N_image_2

24.1.19 Perceptual3DShape

Technique	Perceptual 3D Shape Descriptor
Document	M10324, "Perceptual 3D shape descriptor: Result of core experiment"
Name	In Kyu Park, Hui Zhang, (Multimedia Lab, Samsung Advanced Institute of Technology)
EMail	saitpik@sait.samsung.co.kr, hui.zhang@samsung.com
Type	Application
External Libraries	cv.lib(intel OpenCV) qhull.lib(Geometry Center)
Related Ds/DSs	3D Shape Spectrum Descriptor
Used Ds/DSs	NONE
Input	3D Meshes in VRML2.0 format
Extraction	Yes
Client Appl	Search & Retrieval
Summary	The Perceptual 3D Shape Descriptor provides a compact shape description of 3D data represented as 3D meshes. The Perceptual 3D Shape Descriptor is generated from the part-based representation. It has the form of an attributed relational graph (ARG), composed of nodes and edges. A node represents a meaningful part of the model with unary attributes, while an edge implies binary relations between nodes.
Strong Points	Compactness
Limitations	NONE
Known Problems	NONE
Parameters	bits_per_attribute: the number of bits used for quantizing the descriptor values. Default value is 8 bits.

24.1.20 RegionLocator

Technique	RegionLocator
Document	Text of ISO/IEC CD 15938-3 Multimedia Content Description Interface - Part 3 Visual
Name	Joerg Heuer, Siemens AG
EMail	Joerg.Heuer@mchp.siemens.de
Contact	Joerg.Heuer@mchp.siemens.de
Type	Application
External Libraries	NONE
Related Ds/DSs	NONE
Used Ds/DSs	NONE
Input	Images, parameter file (see ParamRegLocS, ParamRegLocC)
Extraction	Yes
Client Appl	Localization
Summary	This component encodes a polygonal contour representation for localization of objects. The precision of the encoding can be adjusted to the particular needs. There is no other function related to this component.
Strong Points	NONE
Limitations	NONE
Known Problems	NONE
Parameters	NONE

24.1.21 RegionShape

Technique	Region shape descriptor
Document	MPEG-7 Committee Draft Visual, see Region shape descriptor
Name	Whoi-Yul Kim (Hanyang University)
EMail	wykim@email.hanyang.ac.kr
Contact	wykim@email.hanyang.ac.kr
Type	Application
External Libraries	none
Related Ds/DSs	Still region DS
Used Ds/DSs	none
Input	Binary/Gray shape masks
Extraction	Yes
Client Appl	Search & Retrieval
Summary	Shape of an object may consist of either single region or a set of regions as well as some holes in the object in a frame. The region-based shape descriptor describes shape features of the object as long as it can be defined by a shape mask. The mask may be binary to contain holes or multiple disjoint regions within. The mask may also be in gray shade to distinguish different regions within the object. The descriptor consists of a set of 35 quantized coefficients of Angular-Radial Transform of an object.
Strong Points	The descriptor is rotational, scale and position invariant. It can deal with objects with multiple regions including holes in the object. It is robust to noise along the contour of the object that is common during the segmentation process of an image, or to shape deformations due to affine transform. The descriptor is also characterized by its small size, fast extraction time and matching. The data size for this representation is fixed to 17.5byte. The feature extraction and matching processes is straightforward to have low order of computational complexities, and is suitable for tracking shapes in the video data processing.
Limitations	None
Known Problems	None
Parameters	Uses the general XM-SW parameters.

24.1.22 ScalableColor

Technique	Scalable Color
Document	MPEG-7 Committee Draft Visual, see Scalable Color
Name	Aljoscha Smolic (HHI), Jens-Rainer Ohm (RWTH Aachen), Santhana Krishnamachari (Philips)
EMail	smolic@hhi.de, ohm@ient.rwth-aachen.de, santhana.krishnamachari@philips.com
Contact	smolic@hhi.de
Type	Application
External Libraries	none
Related Ds/DSs	Color Space D, Color Quantization D, GoF/GoP Color D, Still Region DS, Moving Region DS
Used Ds/DSs	Color Space D, Color Quantization D
Input	Images
Extraction	Yes
Client Appl	Search & Retrieval
Summary	The Scalable Color descriptor is a color histogram in the HSV color space, which is encoded by a Haar transform. Its binary representation is scalable in terms of bin numbers and bit representation accuracy over a broad range of data rates.
Strong Points	none

Limitations	none
Known Problems	none
Parameters	Uses the general XM-SW parameters.

24.1.23 Spatial2DCoordinates

Technique	Spatial 2D Coordinates descriptor
Document	W3703 MPEG-7 Visual CD, 5.4 Spatial 2D Coordinates
Name	Osamu Hori, Toshimitsu Kaneko, Koichi Masukura (Toshiba R&D Center)
EMail	osamu.hori@toshiba.co.jp, toshimitsu.kaneko@toshiba.co.jp, koichi.masukura@toshiba.co.jp
Contact	koichi.masukura@toshiba.co.jp
Type	Basic Component
External Libraries	NONE
Related Ds/DSs	MotionTrajectoryD, ParametricMotionD, RegionLocatorD, SpatioTemporalLocatorD
Used Ds/DSs	NONE
Extraction	NA
Client Appl	NA
Summary	This component handles mapping data from a default coordinate system to a specific one. This component is referred by other Ds/DSs for describing non default coordinates.
Strong Points	All functions of Spatial2DCoordinates are supported.
Limitations	The maximum number of mumOfMotionParameterSets is 256.
Known Problems	NONE
Parameters	NONE

24.1.24 SpatioTemporalLocator

Technique	Spatio Temporal Locator Extraction Tool
Document	W3703 MPEG-7 Visual CD, 10.2 Spatio Temporal Locator
Name	Osamu Hori, Toshimitsu Kaneko, Koichi Masukura (Toshiba R&D Center)
EMail	osamu.hori@toshiba.co.jp, toshimitsu.kaneko@toshiba.co.jp, koichi.masukura@toshiba.co.jp
Contact	koichi.masukura@toshiba.co.jp
Type	Application
External Libraries	None
Related Ds/DSs	Spatial2DCoordinatesD, MovingRegionDS, SpatioTemporalMaskDS
Used Ds/DSs	RegionLocatorD, TemporalInterporationD
Input	List of Alpha map images (one imege per one frame).
Extraction	Yes
Client Appl	Search Application
Summary	This component generates compressed spatio-temporal region data from alpha-map images. This component supports two trajectory types: the FigureTrajectory and the ParameterTrajectory. The FigureTrajectory describes spatio-temporal regions by trajectories of representative points of refernce region. The ParameterTrajectory describe spatio-temporal regions by trajectories of motion parameters. The TemporalInterpolationD is used for describing trajectories.
Strong Points	Full automatic extracting region data from a sequence of alpth map image.
Limitations	The maximum number of NumberOfReferenceRegions parameter is 10. The maximum number of Vertices in FigureTrajectory is 30.
Known Problems	NONE
Parameters	Trajectory = Select 'FigureTrajectory', 'ParameterTrajectory' or 'MediaTime' to specify trajectory type. StartFrame = Spacify a frame number of a first image of an input image list. EndFrame = Specify the maximum frame number. VerN = Specify a vertices number of a polygon gerenated by the ExtractionTool.

Depth = Specify a value of 'Depth' parameter.

24.1.25 TemporalInterpolation

Technique	TemporalInterpolation ExtractionTool
Document	W3703 MPEG-7 Visual CD, 5.5 Temporal interporation
Name	Osamu Hori, Toshimitsu Kaneko, Koichi Masukura (Toshiba R&D Center)
EMail	osamu.hori@toshiba.co.jp, toshimitsu.kaneko@toshiba.co.jp, koichi.masukura@toshiba.co.jp
Contact	koichi.masukura@toshiba.co.jp
Type	Application
External Libraries	NONE
Related Ds/DSs	MotionTrajectoryD, SpatioTemporalLocatorD
Used Ds/DSs	NONE
Input	Key-point List
Extraction	Yes
Client Appl	NA
Summary	This component genarates temporal interpolating trajectories from a keypoint list. The linier function and quadratic function can be used as interpolating functions.
Strong Points	Users can choose extracting mode: "ErrFix" mode, "KeyPointNumFix" mode and "IntervalFix" mode. The "ErrFix" mode fixes maximum interpolation error. The "KeyPointNumFix" mode fixes total key point number. And the "IntervalFix" mode fixes interval time between keypoints.
Limitations	The maximum number of KeyPointNum is 256.
Known Problems	NONE
Parameters	ExMode = Select "ErrFix", "KeyPointNum" or "IntervalFix" for specify using extracting mode. Degree = Specify the degree of interpolating functions : 1 or 2. ErrLimit = Specify the maximum error limit of "ErrFix" mode. KeyPointNum = Specify the total key point number of "IntervalFix" mode. Interval = Specify the Interval time of "IntervalFix" mode.

24.1.26 TextureBrowsing

Technique	TextureBrowsing
Document	Text of ISO/IEC CD 15938-3 Multimedia Content Description Interface - Part 3 Visual (w4062), Chapter 7.2
Name	yanglim Choi, M/M Lab, Samsung Advanced Institute of Technology
EMail	yanglimc@samsung.com
Type	Application
External Libraries	NONE
Related Ds/DSs	StillRegionDS, StillImageDS, MovingRegionDS
Used Ds/DSs	NONE
Input	Images/Regions(JPEG,BMP,etc)
Extraction	Yes
Client Appl	Search & Retrieval
Summary	This component ?? a very compact texture descriptor for a homogeneously textured Image/Region for browsing using texture features. The descriptor components specifies the regularity, directionality and scale infomation of the texture. The extraction process can be done automatically using Gabor filters as in the XM code or it can also be human annotable.
Strong Points	For quick browsing of textures using the regularity, directionality and scale. Also good for prefiltering process to apply more precise descriptors.
Limitations	NONE
Known Problems	NONE
Parameters	layer ??

24.1.27 TimeSeries

Technique	Time Series
Document	MPEG-7 Visual XM and CD, see "Time Series"
Name	Takehiro FUJITA (Hitachi, Ltd.)
EMail	fujita@crl.hitachi.co.jp
Contact	fujita@crl.hitachi.co.jp
Type	Application
External Libraries	none
Related Ds/DSs	Video Segment DS, Moving Object DS
Used Ds/DSs	Color Layout D
Input	MPEG-Video
Extraction	Yes
Client Appl	Search & Retrieval
Summary	This is designed to assign a temporal series of visual descriptors into a video segment compactly. This can achieve image to video-frame matching and video-frames to video-frames matching functionalities. Two types of TimeSeries are available: RegularTimeSeries (with constant intervals) and IrregularTimeSeries (with various intervals). This library supports the extraction of both types and their matching (see ClientApplication as an example).
Strong Points	High-speed video segment retrieval. Compact representation of a series of descriptors.
Limitations	Compare descriptions of same TimeSeries type for matching.
Known Problems	none
Parameters	TimeIncr: a default interval between descriptors.

24.2 SOUNDTOUCH

SoundTouch is an open-source audio processing library for changing the Tempo, Pitch and Playback Rates of audio streams or files:

- Tempo (time-stretch): Changes the sound to play at faster or slower speed than original, without affecting the sound pitch.
- Pitch (key) : Changes the sound pitch or key, without affecting the sound tempo or speed.
- Playback Rate : Changes both the sound tempo and pitch, as if an LP disc was played at wrong RPM rate.

The SoundTouch library is suited for application developers writing sound processing tools that require tempo/pitch control functionality, or just for playing around with the sound effects. The SoundTouch library Command line interface.

Features:

- Easy-to-use implementation of time-stretch, pitch-shift and sample rate transposing routines.
- High-performance object-oriented C++ implementation.
- Full source codes available for both the SoundTouch library and the example application.
- Clear and easy-to-use programming interface via a single C++ class.
- Supported audio data format : 16Bit integer or 32bit floating point PCM mono/stereo
- Capable of real-time audio stream processing:
  - input/output latency max. ~ 100 ms.
  - Processing 44.1kHz/16bit stereo sound in realtime requires a 133 Mhz Intel Pentium processor or better.
- Platform-independent implementation: The SoundTouch library can be compiled for any processor and OS platform supporting GNU C compiler (gcc) or Visual Studio, for example Win32, Linux, AIX.

- Additional assembler-level and Intel-MMX instruction set optimizations for Intel x86 compatible processors (Win32 & Linux platforms), offering several times increase in the processing performance.
- Compiled executable binaries available for Windows.
- Released under the GNU Lesser General Public License (LGPL).

24.3 Timidity++

TiMidity++ is an open source MIDI to WAVE converter and player. It uses Gravis Ultrasound-compatible patch files and/or SoundFont Banks to generate digital audio data from general MIDI files. The audio data can be played through any sound device or stored on disk. On a fast machine, music can be played in real time. TiMidity++ is written in C and runs under Linux, FreeBSD, HP-UX, SunOS, MacOSX, and Win32, and porting to other systems with gcc should be easy.
Further detailed information and the software to download are available on: http://timidity.s11.xrea.com/index.en.html#links.

Features

- Plays MIDI files without any external MIDI instruments at all
- Understands following formats:
  - SMF (Format 0, 1, 2)
  - MOD
  - RCP, R36, G18, G36 (Recomposer formats)
  - MFi (Version 3; Melody Format for i-Mode)
- Converts MIDI files into various audio file formats:
  - + RIFF WAVE (*.wav)
  - + SUN AU (*.au)
  - + Apple Interchange File Format (*.aiff)
  - + Ogg Vorbis, FLAC, Speex (*.ogg)
  - + MPEG-1 Audio layer 3 (*.mp3) (note: Windows only)
- Uses following formats as digital instrument data
  - Gravis Ultrasound compatible patch files
  - SoundFonts
  - AIFF and WAV data (Some restrictions are there with AIFF/WAV)
- Displays information about the music that is now playing
- Various user interfaces:
  - dumb terminal interface
  - ncurses interface
  - S-Lang interface
  - X Athena Widget interface
  - Tcl/Tk interface
  - Motif interface (runs with lesstif)
  - vt100 interface
  - Emacs front-end (type ``M-x timidity" on your emacs)
  - skin interface: can use WinAmp? skin (Seems not maintained...)
  - GTK+ interface
  - ALSA sequencer interface
  - Windows synthesizer interface
  - Windows GUI interface
  - Windows GUI synthesizer interface
  - PortMIDI synthesizer interface
- Plays remote MIDI files over the network
  - HTTP
  - FTP
  - NetNews

- Plays MIDI files in archive files. Supported formats are:
    - Tar archived (*.tar)
    - Gzip'ed tar (*.tar.gz, *.tgz)
    - Zip compressed (*.zip)
    - LHa compressed lh0, lh1, lh2, lh3, lh4, lh5, lh6, lz4, lzs and lz5 (*.lzh)
- Displays sound spectrogram for the playing music
- Trace playing