AXMEDIS Tutorial on Content Processing

Version 1.4
December 2006

Martin Schmucker – Fraunhofer Institute for Computer Graphics Research IGD, Darmstadt, Germany

Ivan Bruno – Department of Systems and Informatics (DSI-DISIT), University of Florence, Italy

Maulik Sailor – Intelligent Media Systems & Services (IMSS), University Reading, United Kingdom
AXMEDIS attacks the challenges of digital content processing and distribution

Automating Production of Cross Media Content for Multi-channel Distribution
Table of Content

- Tutorial Rationale

- Tutorial Objectives & Outcomes

- AXMEDIS Content Processing (AXCP) – Part I
  - Overview
  - Automation
  - Life-Cycle
  - Core Functionality
  - Extended Functionality
Table of Content (II)

- AXMEDIS Content Processing (AXCP) – Part II
  - AXCP Engine
  - AXCP Grid
  - Usage Scenarios

- Summary, Conclusion and Outlook

- Discussion, Questions and Answers
Tutorial Presentation

Rationale

- Recommended: General Tutorial
- Focus: Automatic Content Processing
  - Underlying basics concepts
  - Usage examples of tools
- Discussion of limits and constraints

Documents are available at http://www.axmedis.org
Tutorial Presentation (II)

- **Intended audience**
  - Decision makers
  - Technical managers
  - Programmers
  - People in the areas of
    - Content production
    - Content distribution and content protection

- **Prerequisites**
  - Basic knowledge of production cycle and tools
  - Basic knowledge of distribution and protection tools
Tutorial Objectives

- Recall overall automatic content processing process
- Overview on individual tools
- AXMEDIS content processing framework usage know-how
- Interaction with tools
- Description of usage scenarios
Tutorial Outcomes

- How to use content processing tools in the AXMEDIS framework
- How to automate content production and processing using AXMEDIS
- Interaction of workflow and content processing
- Distribution of workload by using the AXMEDIS GRID distributed environment
AXMEDIS Tutorial on Content Processing - Part I

- Current Challenges and Key Issues
- AXMEDIS General Overview
- AXMEDIS Content Processing
Current Challenges

- Very fast growing digital-content market

- Major limitations include:
  - Convergence of the media, interoperability of content
  - DRM applications and introduction in several distribution channel
  - Massive processing of content processing and distribution

- Real challenges that are currently being discovered
  - Business-to-Consumer Scenarios (B2C)
  - Business-To-Business Scenarios (B2B)

- Required: innovative means for various scenarios
Key Issue: Flexibility

- Devices and content delivery formats are not static
  - Emerging devices and formats
  - Dynamic market in terms of possibilities and content types and formats

- Required: Flexible Software Tools
  - Support of numerous content types and formats
  - Support of different devices
Key Issue: Automated Processing

- **On-demand distribution:**
  - Production on the basis of requests and profiling (user device, network, etc.), etc.
  - Request depending adaptation and processing

- **Multi-channel distribution**
  - Differing receiving devices
  - Differing distribution modalities
  - Multiple interoperable DRMs, license chain processing

- **Content monitoring**
  - Broadcast channels and networks,
  - Peer-To-Peer networks, Websites, etc.
On Demand Distribution

Content Databases

AXMEDIS Scheduler

AXMEDIS Rule Editor

Few Ready to use rules

User and Device profile

Add info

Activate Rule

Personalized Content on Demand

AXCP GRID

Content: Search, Selection, Acquisition, Production, Adaptation, Transcoding, Formatting, Packaging, Protection, Publication and Licensing on Demand

Distributor front end Server

players

AXMEDIS CS

AXMEDIS PMS

Content Databases

AXMEDIS Scheduler
Multichannel Distribution

AXMEDIS Content Processing Tools

License distribution and player/device verification and supervision

Distribution Server Front-end

License distribution and player/device verification and supervision

Distribution Server Front-end

Augmented License

Content un-protecting and rights exploitation

Ask for the License

Users and devices

Open Licensing Model

Content un-protecting and rights exploitation

Get the content
Different channels

Different DRMs
- AXMEDIS
- OMA
- Windows Media

Collection of Actions Logs Records
AXMEDIS Advantages

- **Flexible and innovative solutions for**
  - Content production and management
  - Content distribution and aggregation
  - Digital Rights Management (DRM)

- **Cost Efficiency through underlying principles**
  - Automatic massive content production and processing
  - GRID technology
  - Extensibility through plug-ins technology

- **Cost Efficiency through the integration with**
  - Existing Content Management Systems (CMS)
  - Existing e-Service, e.g. back-office and workflow support
AXMEDIS Advantages (II)

- **AXMEDIS Automated Content Processing**
  - Massive and small scale processing
  - Locally performed or Workflow controlled
  - On any kind of Digital Resource not only AXMEDIS objects

- **AXCP Applications for massive processing as**
  - Production/packaging platform for producers and integrators
  - Protection of objects, and protection information processing
  - Transcoding/adaptation platform for distributors
  - License Production, or as License Sever/processor
  - etc.
Summary: AXMEDIS

- Intelligent and flexible solution
  - Automated content processing and production
  - Efficient usage of available resources
  - Interoperability with existing infrastructure
  - Automatic provision of new service
Content Processing With AXMEDIS

Automation of content processing
AXMEDIS General Overview

- AXMEDIS Editor
- AXMEDIS Automatic Content Processing
- AXMEDIS Architecture
- AXMEDIS Content Processing
AXMEDIS Editor

- manual production of AXMEDIS objects
- inspection of automatically produced objects
- finishing AXMEDIS objects pre-produced automatically

AXMEDIS Tutorial on Content Production,

LT1: Tuesday, 12th (yesterday)
AXMEDIS Editor

- It integrates many Editors & Viewers to handle all the aspects of the AXMEDIS Objects production
  - Resource
  - Metadata
  - DRM
  - Protection
  - Presentation
  - Workflow
  - Annotation
  - …

- AXMEDIS Tutorial on Content Production,
  - LT1: Tuesday, 12th (yesterday)
Ringtone Adaptation Plug-in

- Adaptation of Ringtones of popular formats

- Transcoding on-demand.

- Main functions
  - Convert to different formats
    - MP3, Wav, etc
  - Resample the Frequency, Bit rate etc
  - Clip to 5 sec, 30 sec clip etc
Convert
Convert to MP3

Ringtone Adaptation: `convert_to_MP3(InputResource, OutputResource)`

**Parameters**

- **in InputResource:RESOURCE**
  - Resource [Test1.wma]
  - The Resource to be converted

- **out OutputResource:RESOURCE**
  - New Resource
  - Where the produced resource will be stored

**Result**

- **result:STRING**
  - SUCCESS

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

[Buttons: Execute, Close]
Convert to Wav

RingtoneAdaptation: convert_to_WAV(InputResource, OutputResource)

Parameters

- in InputResource:RESOURCE
  - Resource [Test1.wmi]: The Resource to be converted
- out OutputResource:RESOURCE
  - New Resource: Where the produced resource will be stored

Result

- result:STRING
  - SUCCESS

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

Execute  Close
Resample

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>in InputResource:RESOURCE</td>
<td>Resource[]</td>
<td>The Resource to be converted</td>
</tr>
<tr>
<td>out OutputResource:RESOURCE</td>
<td>New Resource</td>
<td>Where the produced resource will be stored</td>
</tr>
<tr>
<td>in OutputSamplingRate:UINT32</td>
<td>44100</td>
<td>Sampling rate of the output audio file (default: sampling rate of the input)</td>
</tr>
<tr>
<td>in OutputNumChannels:UINT16</td>
<td>2</td>
<td>Number of channels of the output audio file (default: number of channels of the input)</td>
</tr>
<tr>
<td>in OutputBitRate:UINT16</td>
<td>128</td>
<td>Bit rate of the output audio file - Only applies to compressed audio file formats (default: 64 kb)</td>
</tr>
</tbody>
</table>

**Result**

- result:STRING: **SUCCESS**

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

[Image of the Resample interface with parameters and result]
Convert & Resample

![Image of Convert & Resample window showing parameters and results]

**Parameters**

- **in InputResource**: RESOURCE [Test1.wav]
  - The Resource to be converted and resampled.
- **in Mimetype**: STRING audio/x-mpeg
  - Mimtype for output resource.
- **out OutputResource**: RESOURCE New Resource
  - Where the produced resource will be stored.
- **in OutputSamplingRate**: UINT32 44100
  - Sampling rate of the output audio file (default: sampling rate of the input).
- **in OutputNumChannels**: UINT16 2
  - Number of channels of the output audio file (default: number of channels of the input).
- **in OutputBitRate**: UINT16 128
  - Bit rate of the output audio file - Only applies to compressed audio file formats (default: 64 kb).

**Result**

- **result**: STRING SUCCESS
  - The result of import, SUCCESS if ok, ERROR followed by a message in case of error.

**Buttons**

- **Execute**
- **Close**
GetInfo

Ringtone Adaptation: getInfo(InputResource, SamplingRate, NumChannels, BitRate, Duration)

Parameters:
- **in InputResource**: RESOURCE
- **out SamplingRate**: UINT32
- **out NumChannels**: UINT16
- **out BitRate**: UINT16
- **out Duration**: STRING

- **Resource [Test1.wmx]**
- **44100**
- **2**
- **128**
- **0:2:5:1**

The Resource to be converted
Sampling rate of the input ring tone
Number of channels of the input ring tone
Bit rate of the input ring tone - (default: 64 kb)
Duration of the Ringtone File

Result:
- **result: STRING**: SUCCESS

The result of the operation, SUCCESS if ok, ERROR followed by a message in case of error

[Buttons: Execute, Close]
Clip


**Parameters**

- **in InputResource:RESOURCE**
  - Value: Resource [Test1.wav]
  - Description: The Resource to be converted

- **out OutputResource:RESOURCE**
  - Value: New Resource
  - Description: Where the produced resource will be stored

- **in Mimetype:STRING**
  - Value: audio/x-mpeg
  - Description: MimETYPE for output resource

- **in ReadStartingTime:FLOAT**
  - Value: 0.0
  - Description: Starting time for the clip (default: beginning of the file)

- **in ReadEndingTime:FLOAT**
  - Value: 30.0
  - Description: Ending time for the clip (default: end of the file)

**Result**

- **result:STRING**
  - Value: SUCCESS

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

**Buttons:**
- Execute
- Close
AXMEDIS Editor

- AXMEDIS Tutorial on Content Production,
  ➜ LT1: Tuesday, December, 12th, 2006 (yesterday)
AXMEDIS Automated Content Processing Tools

Available functionalities in the AXMEDIS Framework
AXMEDIS Automated Content Processing Area

- Properties
- Architecture
- AXMEDIS Content Processing Engine
- AXMEDIS GRID
Automatic Content Processing

- Content distribution and production characteristics:
  - Content lifetime
  - Demand is time dependent
  - …

- Example:
  - Creating a content for a current event
  - Now: winter scenario … started to snow (… or not)
  - Manual production is (too) time consuming

- How can the AXMEDIS Automatic Content Processing (AXCP) Area support you?
  - Scalability
  - Extensibility
  - Essential functionality to fulfil real world’s requirements
AXCP in the AXMEDIS Architecture

AXMEDIS Factory
- AXCP Editors
- AXMEDIS database Area
- AXMEDIS Content Processing Engines and Scheduler GRIDs
- AXMEDIS Accounting area
- Programme and Publication

Workflow Management tools

AXEPTool Area
- AXEPTools
- AXMEDIS Factory
- AXEPTools
- AXEPTools

CMSs
- Crawlers

Protection and Supervising tools
- AXMEDIS Certifier and Supervisor
- Protection Manager Support

AXMEDIS Portal

AXMEDIS Players
- Distributor
- Distributor

AXMEDIS Network
AXMEDIS Content Processing GRID

AXMEDIS Scheduler

Workflow manager

AXMEDIS Rule Editor

Your AXCP Rules

Any Plug-in for content processing

AXCP nodes

WS, FTP, etc.

Your CMSs

AXMEDIS Database

Distribution Channels and servers
AXMEDIS Automated Content Processing Capabilities

- **Automated Content and Metadata Retrieval**
  - Content and Metadata Ingestion and Gathering
  - Content Query, Retrieval and Storage

- **Automated Content and Metadata Processing**
  - Content Adaptation and Transcoding
  - Metadata Generation and Mapping
  - Content Composition and Formatting
  - Content Protection and Licensing

- **Automated Content Distribution**
  - Content Publication
  - Content Distribution
  - Profile management and processing
  - Production of Content on Demand
Content production, protection and distribution

AXMEDIS database

Composing

Formatting

Fingerprint

Adaptation

Protection

Fingerprint Extraction, Descriptor Extraction, etc.

Adaptation of Content, Metadata, DRM, etc.

Object Protection, Encoding, License gen, Governed Object Gen

Your CMSs

Automatic Content Crawling and AXMEDIS Component Production

Distribution channels

Integrators

Providers

P2P B2B network

Providers
Content and Metadata Retrieval

- **Access to numerous existing DBs and CMSs for**
  - Content and Metadata Ingestion and Gathering
  - Content Query, Retrieval and Storage
  - Automatic migration of digital contents

- **Access to several different resources**
  - File Systems: Win, Linux, MAC, etc.
  - ODBC, JDBC, etc.
  - Native DB: DB2, Oracle, MS-SQL, MySQL, etc.
  - Protocols: IMAP, POP, Z39.50, etc.
  - XML databases

- **Automation of**
  - Loading and saving of AXMEDIS objects
Content and Metadata Processing

- **Metadata Generation**
  - Calculation of Low and High Level Descriptors
  - For AXMEDIS objects and included resources

- **Metadata Mapping**
  - Managing generic, AXInfo and DublinCore Metadata

- **Content Composition**
  - Putting together content different kinds of raw assets to create a new digital item
    → AXMEDIS Object (or MPEG-21 format)
  - Composing different AXMEDIS objects selected from the AXMEDIS database
Content and Metadata Processing (II)

- Content Adaptation and Transcoding
  - Adaptation of content (digital item adaptation, DIA)
  - For distribution via different channels to users’ platforms such as i-TV, mobile, PC, etc...

- Content Formatting
  - Modifying digital resources according to a specific format
    - File format and properties
  - Applying a (content dependent) formatting style
    - graphic layout
    - spatial constraints
    - quality limitations
    - synchronization between audio and images, etc.
Content and Metadata Processing (III)

- **Content Protection**
  - Preventing un-authorized content access
  - Preventing un-authorized content manipulations

- **Content Licensing**
  - DRM and License production
  - DRM and Licence processing
  - Offline and on-demand services
Content Distribution

- Content Publication
  - metadata adaptation and mapping
  - publication of AXMEDIS objects on external channels in the B2B distribution (AXEPTools)

- Content Distribution
  - AXMEDIS objects distribution via available channels in the B2C scenarios

- Profile management and processing
  - Considering the characteristics of transmission channels and receiving device

- Production of Content on Demand
  - Creating new content upon customer’s request
Extensibility

- Provided functionality is extensible

- Plugin Interface
  - To include further existing tools
  - To include future tools

- Plugin SDK (Software Development Kit)
  - For easy integration of your algorithms as plugins
  - Including documentation and source code examples
AXMEDIIS Content Processing GRID

What?

How?

AXMEDIS Database

AXMEDIS Rule Editor

Workflow manager

AXCP Scheduler

AXCP nodes

Any Plug-in for content processing

Your AXCP Rules

Your CMSs

WS, FTP, etc.

Distribution Channels and servers

Any Plug-in for content processing
How can you automate content processing within AXMEDIS?

Definition of rules for automated content processing
Rules are expressed in a scripting language

- What is a rule?
- What can be done with a rule?
- How to write, test and finalize a rule?
- How to execute a rule?
Rules within AXMEDIS

- An AXCP Rule is a scripted procedure:

\[ R = f(S_1, S_2, ..., S_m, P_1, ..., P_m) \]

Where:
- \( S_i \) is a database Selection
  - It is a sequence of queries to be sent to the AXMEDIS Database to retrieve objects IDs;
- \( P_i \) is a parameter (basic type as integer, string, Boolean, etc.);
  - For example, coordinates for a formatting, size or value of object collection to be created, destination of the objects, name of the author, etc.
- \( f \) is the identifier of rule (name of rule or an ID);
- \( R \) is the result of the rule application.
  - It can be a new AXMEDIS object, or a metadata manipulation, the license of an AXMEDIS object, a message to be returned to invokers, etc...
AXCP Rule XML description

- General metadata regarding: rule name, AXRID (rule identifier), rule version, rule type, software name, version of software, date of production, time of production, author, affiliation, URL, comment, last modification and terminal ID. *(Header)*

- Temporal metadata describing conditions for firing the rule, expiration time, periodicity and the rule status (“active” or “inactive”) and *(Schedule)*

- List of arguments (parameters and selections), list of dependences (required AXMEDIS plug-ins) and the rule body (the script code to run). *(Definition)*
AXCP Script Language

- AXCP Rules are formally based on a script language
- Basis: Javascript Language (Spidermonkey by Mozilla)
- Extended functionality that is provided by the AXMEDIS framework for managing
  - AXMEDIS Objects, Digital Resources
  - License, Metadata
  - Network, Database Access and Web Service
  - Profiling (user, device, context)
  - ...
- Additional functions/utilities
  - File System management
  - Invoking/Execution of external tools
  - MimeType management
  - Zip/Unzip Archive

⇒ Everything that can be done manually with AXEditor can be done automatically with the AXCP

⇒ … and more 😊
AXCP Tools: Rule Editor

Where and how to write, test and execute rules
AXCP Rule Editor

- IDE (integrated development environment) for creating, testing and managing AXCP rules

- Provides a set of tools and views to help the user during the editing and building of rule.

- Integrates the AXCP rule executor in order to provide functionalities for:
  - executing, debugging, testing and validating

- Provides an online help
  - Documentation about script functions and classes of the AXCP Script Language.
  - List of AXCP Plugins and documentation about exposed functionalities
AXMEDIS Content Processing Area: AXCP Rule Editor

- It is an IDE tool for:
- Creating and editing AXCP Rules
- Defining parameters and required AXMEDIS Plugins
- Editing, checking and debugging JS code
- Activating AXCP Rules into the AXCP Rule Engine
AXMEDI’S Content Processing GRID

- AXMEDIS Rule Editor
- Workflow manager
- AXCP Scheduler
- AXCP nodes
- Your CMSs
- AXMEDIS Database
- Distribution Channels and servers
- Your AXCP Rules
- Any Plug-in for content processing

Examples

WS, FTP, etc.
Examples

Using the Available Functionality for Automated Content Processing
Example 1 of the User Manual DE5.0.1.1

Goal:
- Create a simple script for resizing an image resource named AXMEDIS_logo.png stored in the C:\\ path and saving the new resized image on the disk.

Method:
- Writing and debugging the script
- Comparing the original and new to see the result
Steps

1. Create an empty digital resource (image)
2. Load the image file into the resource
3. Use the Image Processing plug-in for scaling the image resource
4. Saving the scaled image resource
Example 2 of the User Manual DE5.0.1.1

Goals:
- Create a simple script for resizing an image resource stored on file system and saving the new resized image on the disk.
- Generalize the previous script inserting some arguments in the script.
  - Location for loading and saving the image
  - Resulting image dimensions
  - Output image format
- Obtain a rule that will remain valid and will be not modified in the future.

Method:
- Writing and debugging the script
Steps

Parameter definition

1. Definition of rules arguments (*input_path*, *output_path*, *width*, *height*, *out_mime_type*)

Parameter usage

2. Create an empty resource

3. Load the image file by the *input_path* argument

4. Use the Image Processing plugin for *scaling* the image

5. Use the Image Processing plugin for *conversion* the image

6. Resulting image is saved in the location specified by the *output_path* argument
Creating AXMEDIS Objects

- Example 3 of the User Manual DE5.0.1.1

- Goal:
  - Example 2 (Loading manipulating image resource)
  - Converting the resource into a different format
  - Create an AXMEDIS object
  - Filling metadata
  - Embedding an image resource
  - Storing on disk and/or uploading on DB

- Method
  - Write and debugging the script
  - Show of the produced AXMEDIS Object with AXEditor
Steps:

1. Definition of rules arguments (*input_path*, *output_path*, *width*, *height*, *out_mime_type*)

2. Create an empty resource

3. Load the image file by the *input_path* argument

4. Use the Image Processing plugin for *scaling* the image

5. Use the Image Processing plugin for *conversion* the image

6. Creating an empty AXMEDIS object

7. Adding meta data and the image resource

8. Save the AXMEDIS object

9. Open the AXEditor for visualisation
“Full Round” AXMEDIS Automated Object Production

■ Steps:
  ▶ Example 3 (previous example: AXObject creation)
    • Filling metadata
    • Embedding a resource
    • Adapting the resource into a different format
  ▶ Registering Object
  ▶ Protecting Object
  ▶ Creating and storing licenses
  ▶ Storing on disk and/or uploading on DB

■ Method
  ▶ Write and debugging the script
  ▶ Show of the produced AXMEDIS Object with AXEditor
Steps:

1. Setting of the license related parameters
2. Create an empty resource and AXMEDIS object
3. Load the resource
4. Adding meta data and the resource in the AXMEDIS object
5. Save the AXMEDIS object on disk and AXMEDIS database
6. Creating licences for distribution
7. Creating license for the end user
8. Fruition – Using the AXMEDIS object
Automated Content Processing

What you have seen
- What is a rule and what can be done with it?
- Writing a rule in the AXCP Rule Editor for automated content processing?
- How to execute and testing a rule in the AXCP Rule Editor?

Next:
- Extended functionalities for content processing
Which further functionalities are available?

Extended functionalities provided by the AXMEDIS Framework for Automated Content Processing
Examples: Extended Functionalities

- Content Description and Meta Data Creation
- Meta Data Mapping
- Adaptation and Transcoding
- Content Authentication
Content Description and Meta Data Creation

- **Automatic extraction/calculation of**
  - Low Level Descriptors (LLDs)
  - Perceptual Hashes/Content Fingerprints
  - High Level Descriptors (LLDs)

- **List of available functionalities**
  - Audio - LLDs
    - MPEG-7 AudioWaveform
    - MPEG-7 AudioPower
    - MPEG-7 AudioSpectrumEnvelope
    - MPEG-7 AudioSpectrumCentroid
    - MPEG-7 AudioSpectrumSpread
    - MPEG-7 AudioSpectrumFlatness
    - Mel-Frequency Cepstrum Coefficients
Content Description and Meta Data Creation (II)

- Continued list of available functionalities
  - Audio – HLDs
    - Rhythm
    - Audio File Segmentation
    - Music Genre
  - Video
    - Basic Integration of MPEG-7 reference implementation
  - Text
    - Language Guesser
    - Keyword Extraction
Demonstration: Text processing

- **Document Adaptation (DA)**
  - transcoding between various text formats (PDF, HTML, Postscript, RTF, plain text)

- **Language guesser (LG)**
  - Retrieving the main language from a plain text document

- **Keyword extractor (KE)**
  - Extraction of mono-term and multi-term keywords from an plain text (English, Italian, German, Spanish and French, only English is implemented so far)

- Automatically extracted descriptors can be used to fill AXMEDIS objects metadata, allowing better search results.
Text Processing

PDF document

AxObject

PDF

DA plug-in

AxObject

LG plug-in

AxObject

English

PDF

HTML

TXT

KE plug-in

AxObject

English

{plagiarism detection, document structure, tree structure, level of the document}

PDF

HTML

TXT

PDF

HTML

TXT
Steps

1. Loading resource

2. Transcoding to HTML

3. Embedding the HTML resource in the object

4. Transcoding to plain text

5. Embedding the plain text resource in the object
Plagiarism detection plug-in

- Detection of plagiarism behaviour: plagiarism between two textual documents

- Basic model: a set of modifications (insertion, deletion, substitution) performed

- Return value: normalised similarity value

- Can be combined with other tools, e.g. keyword extractor
Plagiarism plugin usage scenarios

External documents or axmedis documents to be compared

Document Base

Candidate Document

Keyword comparison

Topic sharing documents

Plagiarism plugin comparison

Plagiarized documents
Steps

1. Defining the reference document
2. Defining the documents to compare with
3. Comparing the documents
Adaptation and Transcoding

- Transcoding to and from numerous formats

- Supported media types:
  - Audio
  - Images
  - Text
  - Video

- Metadata Adaptation

- Rights Information Adaptation
  - Licenses: MPEG-21 ↔ OMA

- Dynamic Transcoding
  - See: Workflow Tutorial, today, starting 2pm
Audio Adaptation

Goal:
- Genre detection (only possible with WAVs)

Steps
- Loading the input resource (MP3 format)
- Converting the input resource to WAV
- Apply a genre detection
- Output: genre
Meta Data Mapping

- **Manages**
  - Generic, AXInfo and DublinCore Metadata

- **Provides functionalities for**
  - Extracting Metadata from an AXMEDIS Object
    - Adapting Metadata
      - Loading adaptation style sheet
      - Transforming extracted metadata
    - Embedding the transformed metadata into an AXMEDIS Object
Steps

1. Open AXMEDIS object
2. Initialize the Metadata Mapper
3. Extract the Metadata from the AXMEDIS object
4. Load the saved XSLT file from the disk
5. Transform the metadata producing the new Metadata
6. Embed the new metadata into an AXMEDIS Object
7. Save the AXMEDIS Object
Content Authentication

- Cryptographic Hash Functions
  - All kind of content

- Perceptual Hash Functions
  - Audio
  - Images
  - Video
Further Functionalities

- I/O functions
- File and directory access
- Profiling of devices, distribution channels and users
- Network access: FTP, HTTP, ODBC, webservices, ...
Summary

- Why to automate content processing processes
- Automated content processing in AXMEDIS
  - Writing, executing and debugging rules
  - Available functionality
  - Examples for using the available functionality
AXMEDIS Tutorial on Content Processing - Part II

- AXCP Rule Scheduler
- AXCP GRID
- Complex Scenarios
- Summary and Conclusion
- Discussion, Questions and Answers
AXMEDIIS Content Processing GRID

Rule Execution in GRID

Workflow manager

AXMEDIS Rule Editor

AXCP Scheduler

AXCP nodes

Your CMSs

AXMEDIS Database

Your AXCP Rules

Any Plug-in for content processing

WS, FTP, etc.
AXCP Rule Engine

Managing the Rule Execution in the AXMEDIS Grid
AXCP Rule Engine

- **Consists of**
  - Rule Scheduler (Server Side)
  - Rule Remote Executor (Client Side): AXCP GRID Node

- **Rule Scheduler: Internal Scheduler and Dispatcher for**
  - rule installation
  - rule firing
  - rule executor discovering and management
  - rules scheduling and dispatching according to the executor profile
  - communication with the AXMEDIS environment (workflow)
  - ...

- **Rule Remote Executor**
  - Consisting of the same JavaScript engine
AXCP Scheduler: Functionalities

- Install & Activate a Rule in AXCP Grid
- Run a Rule in AXCP Grid
- Deactivate a Rule
- Suspend a Rule
- Pause a running Rule
- Resume a previously suspended Rule
- Kill a Rule immediately
- Remove a Rule from the AXCP Grid
- Determine the Status for any Rule in the Grid
- Retrieve the Logs for any Rule in the Grid
- Retrieve the List of Rules in the Grid
- Retrieve a particular Rule in the Grid

→ manually or with a remote client (e.g. workflow)
Remote Rule Executor

- **Application**
  - Standalone

- **Node of the GRID**

- **Properties**
  - CPU Monitoring
  - CPU Workload Constraints
  - Communication with the Scheduler (GRID Node)
AXCP GRID Node

- Workload setting (see configuration.xml)

```xml
<Module category="" id="WORKLOAD_SETTINGS">
    <Parameter name="MON" type="string">
        0;0;0;0;0;30;50;30;20;30;45;50;20;30;45;50;30;30;0;0;0;0
    </Parameter>
    <Parameter name="TUE" type="string">
        0;0;0;0;0;30;30;30;30;20;30;45;50;20;30;45;50;30;30;0;0;0;0
    </Parameter>
    <Parameter name="WED"  type="string">
        0;0;0;0;0;30;30;50;60;20;30;45;50;20;30;45;50;30;30;0;0;0;0
    </Parameter>
    <Parameter name="THU" type="string">
        0;0;0;0;0;60;60;60;60;20;30;45;50;20;30;45;50;30;30;0;0;0;0
    </Parameter>
    <Parameter name="FRI" type="string">
        0;0;0;0;0;50;50;50;50;20;30;45;50;20;30;45;50;30;30;0;0;0;0
    </Parameter>
    <Parameter name="SAT" type="string">
        0;0;0;0;0;50;50;50;50;20;30;45;50;20;30;45;50;30;30;0;0;0;0
    </Parameter>
    <Parameter name="SUN" type="string">
        0;0;0;0;0;50;50;30;70;20;30;45;50;20;30;45;50;30;30;0;0;0;0
    </Parameter>
</Module>
```
Running Rules with the Workflow

 AXMEDIS Factory

 AXMEDIS Content Processing Engines and Scheduler GRIDs

 AXMEDIS databases

 Crawlers

 CMSs

 AXMEDIS Accounting area

 Programme and Publication

 Protection and Supervising tools

 AXMEDIS Certifier and Supervisor

 Protection Manager Support

 AXMEDIS Portal

 AXMEDIS Players

 AXMEDIS Network

 AXMEDIS Factory

 AXMEDIS Network

 AXMEDIS Players

 AXMEDIS Network

 AXMEDIS Players
AXMEDIS The Protection and Control Process

AXMEDIS Certifier and Supervisor
AXMEDIS Protection Manager
AXMEDIS Compliant Players
Any Distributor Front End Server
Payment Front end Server

New Protected Object: Post ProtInfo (WS)
do license (WS)

AXCP

Metadata Resource
GRID in action

- Put in Execution the Scheduler
- Put in Execution the single GRID Nodes
- Joining of nodes to the GRID
- On the AXCP Editor: Installing Rules
- On the Scheduler: Running Rules with the Workflow
Discovered AXCP GRID Node

Collegamento a axcpgridnode.exe

***************************************************************************
* Welcome on AXMEDIS AXCP GRID NETWORK *
***************************************************************************

Initializing AXCP GRID NODE
-> Starting GRID peer functions...
-> GRID peer functions started
-> GRID Cpu Monitor started
AXCP GRID NODE Ready
Connected to AXCP Scheduler (192.168.0.191)
Profile request by 192.168.0.191
Profile sent to 192.168.0.191
Installing Rules

- **Manually using the scheduler**

- **Installing rules with the AXCP Editor**
  - Script for Automated Production of content and production of licenses

- **Activating rule using external application**
  - AXCP Rule Editor, Workflow, …
  - Workflow is activating them

- **Running a rule**
  - On-demand execution
Massive Production of AXObjects

Distributed workload

- Steps
  1. Loading raw resources from file system
  2. Content adaptation
  3. Content protection
  4. Posting protection information on AXCS
  5. DRM Licensing:
     1. Production of Mother license for distribution
     2. Posting of licenses on the PMS
     3. Production of some final user licenses only for adapted objects
     4. Posting of licenses on the PMS
  6. Distribution

- Results
  1. Master Version (protected and unprotected)
  2. Adapted Objects (protected and unprotected)
What is produced

- Some Different GRID nodes
  - only two of them have capabilities to manage the rules in terms of plugins installed

- Two rules activated on two different nodes

- The rules work on some resources
  - a PDF is adapted to produce two objects in TXT and HTML
  - From an MP3 is produced an object
  - A wav file is adapted to produce an object with an MP3
  - From images in different formats, some adaptation resizing are performed

- More than 30 AXMEDIS objects are created
AXMEDIS The Protection and Control Process

AXMEDIS The Protection and Control Process

Any Distributor
Front End Server

Payment
Front end Server

AXMEDIS
Protection Manager

AXMEDIS
Certifier and Supervisor

New Protected
Object: Post
ProtInfo (WS)

do license (WS)

AXMEDIS Compliant Players

AXCP

Metadata
Resource

Metadata
Resource

Any Distributor
Front End Server
Complex Scenarios

Real Scenario taken from Accademia Nazionale di Santa Cecilia
AXMEDIIS Content Processing GRID

Complex Scenarios

Workflow manager

AXMEDIIS Rule Editor

AXCP Scheduler

AXCP nodes

Your AXCP Rules

Any Plug-in for content processing

WS, FTP, etc.

Your CMSs

AXMEDIIS Database

Distribution Channels and servers

AXCP nodes
Complex Scenarios


- Examples:
  - Content gathering from file system
  - Crawling from CMS using SearchBox Tool
  - Automatic SMIL representation
Content gathering from file system

HTML + ... + custom metadata + CSS

AXMEDIS AXCP Rule

Axmedis Object: Resources (HTML, Images, mp3, CSS, ..) metadata ...
Crawling from CMS using SearchBox Tool

ANSC CMS

EAD XML

Searchbox

http get

custom metadata

AXMEDIS AXCP Rule

Axmedis Object: Resources + metadata
Automatic SMIL Representation

ANSC CMS

EAD XML

Searchbox

AXMEDIS AXCP Rule

Axmedis Object with SMIL

http get

custom metadata

SMIL template
Summary

- Rules Definition, test and validation
- Rules installation and Execution
- Rules Scheduling
- AXMEDIS GRID management
Adding Your Plugins

- Flexible Plugin Interface already used to integrate the extended functionality

- Examples:
  - ImageMagick,
  - FFMPEG,
  - CryptLib,
  - MPEG-7 reference implementation,
  - ...

- External tools
  - M2ANY: AudioID (integrated)
  - ISHCE: Audio Watermarking (under development)
The AXMEDIS Scripting Language provides a flexible solution for automatic content processing.

The AXMEDIS Core Functionality provides the basic method for the management of digital content and corresponding rights.

External Functionality included in AXMEDIS enriches the basic functionality to cover daily tasks.

The AXMEDIS GRID allows a dynamic load balancing.

The extensibility of the AXMEDIS Framework provides the flexibility for future scenarios:
  ▶ scripting and creation of new AXMEDIS plugins.
Conclusions

- AXMEDIS reduces costs for content management by providing a solution for automating the content processing, production, protection and distribution.

- AXMEDIS reduces distribution and aggregation costs in order to increase accessibility with a
  - Peer-to-Peer platform at Business-to-Business level
  - Integration of content management systems
  - Integration of workflows
Contact

- **Martin Schmucker,**
  Fraunhofer Institute for Computer Graphics Research,
  Darmstadt, Germany
  Web: [http://www.igd.fraunhofer.de/igd-a8](http://www.igd.fraunhofer.de/igd-a8)

- **Prof. Paolo Nesi, Ph.D.**
  DISIT-DSI
  Department of Systems and Informatics
  Distributed Systems and Internet Technology Lab
  University of Florence
  Via S. Marta 3, 50139 Firenze, Italy
  Email: nesi@dsi.unifi.it
  Web: [http://www.AXMEDIS.org](http://www.AXMEDIS.org)
Lunch Break
Steps

1. Open AXMEDIS object
2. Initialize the Metadata Mapper
3. Extract the Metadata from the AXMEDIS object
4. Load the saved XSLT file from the disk
5. Transform the metadata producing the new Metadata
6. Embed the new metadata into an AXMEDIS Object
7. Save the AXMEDIS Object
AXMEDIS Tutorial on Content Processing

Version 1.4
December 2006

Martin Schmucker – Fraunhofer Institute for Computer Graphics Research IGD, Darmstadt, Germany

Ivan Bruno – Department of Systems and Informatics (DSI-DISIT), University of Florence, Italy

Maulik Sailor – Intelligent Media Systems & Services (IMSS), University Reading, United Kingdom