



MAD



Content Analysis Tools

Deliverable 15.4

DOCUMENT IDENTIFIER PS_WP15_ D15.4_CA_Tools_v1.01

DATE 2007-02-16

ABSTRACT This document describes the Content Analysis Tools delivered to the PrestoSpace Factory subsystem developed by the Metadata Access and Delivery Area. The tools described here are part of the MAD Documentation Unit. This document is a part of a two-pieces deliverable, which includes in addition the software packages delivered as separate objects.

AUTHOR (COMPANY) Alberto Messina (RAI), Werner Bailer, Peter Schallauer (JRS), Valentin Tablan (University of Sheffield), Roberto Borgotallo (RAI), Daniele Airola (RAI), Laurent Boch (RAI)

EDITOR Alberto Messina (RAI)

VERSION 1.01

DISSEMINATION PU : Public

DOCUMENT HISTORY

Release	Date	Reason of change	Status	Distribution
0.01	2006-04-10	First Draft	Draft	Confidential
0.02	2006-04-11	Second Draft	Draft	Confidential
0.03	2006-11-14	Paragraphs rearrangements	Draft	Confidential
0.04	2006-11-17	Drafting RAI's parts	Draft	Confidential
0.05	2007-01-10	Added text to RAI's parts	Draft	Confidential
0.10	2007-01-12	Added first draft of JRS parts	Draft	Confidential
0.20	2007-01-15	Added first draft of SHEF parts	Draft	Confidential
0.50	2007-01-18	First complete draft	Draft	Confidential
0.52	2007-02-13	Added licensing conditions for some tools. First complete review	Draft	Confidential
0.99	2007-02-13	Final draft sent out to MAD for completion and final comments.	Draft	Confidential
1.00	2007-02-16	Final	Final	Confidential
1.01	2008-02-26	Dissemination level changed to Public	Final	Public

Contents

- [**Contents..... 2**](#)
- [**1 Overview..... 4**](#)
 - [1.1 Scope of the document..... 4](#)
 - [1.2 Selection Criteria..... 4](#)
- [**2 Target Architecture..... 4**](#)
- [**3 Supported Tools..... 5**](#)
 - [3.1 Shot Boundary Detection..... 5](#)
 - [3.1.1 Description..... 5](#)
 - [3.1.2 Rationale for Selection..... 6](#)
 - [3.1.3 Integration in Documentation Platform..... 6](#)
 - [3.2 Key Frame Detection..... 7](#)
 - [3.2.1 Description..... 7](#)
 - [3.2.2 Rationale for Selection..... 8](#)
 - [3.2.3 Integration in Documentation Platform..... 8](#)
 - [3.3 Stripe Image Extraction..... 9](#)
 - [3.3.1 Description..... 9](#)
 - [3.3.2 Rationale for Selection..... 9](#)
 - [3.3.3 Integration in Documentation Platform..... 9](#)
 - [3.4 Camera Motion Detection..... 10](#)
 - [3.4.1 Description..... 10](#)
 - [3.4.2 Rationale for Selection..... 10](#)
 - [3.4.3 Integration in Documentation Platform..... 10](#)
 - [3.5 Visual Activity..... 11](#)
 - [3.5.1 Description..... 11](#)
 - [3.5.2 Rationale for Selection..... 11](#)
 - [3.5.3 Integration in Documentation Platform..... 11](#)
 - [3.6 Speech to Text Transcription 12](#)
 - [3.6.1 Description..... 12](#)
 - [3.6.2 Rationale for Selection..... 12](#)
 - [3.6.3 Integration in Documentation Platform..... 12](#)
 - [3.7 Audio Structuring and Segmentation..... 13](#)
 - [3.7.1 Description..... 13](#)
 - [3.7.2 Rationale for Selection..... 14](#)
 - [3.7.3 Integration in Documentation Platform..... 14](#)
 - [3.8 Multimedia Structure Detection..... 15](#)
 - [3.8.1 Description..... 15](#)
 - [3.8.2 Rationale for Selection..... 15](#)
 - [3.8.3 Integration in the Documentation Platform..... 15](#)

- 3.9 Editorial Parts Segmentation..... 16
 - 3.9.1 Description..... 16
 - 3.9.2 Rationale for Selection..... 16
 - 3.9.3 Integration in Documentation Platform..... 16
- 3.10 Reference Video clip detection..... 18
 - 3.10.1 Description..... 18
 - 3.10.2 Rationale for Selection..... 18
 - 3.10.3 Integration in Documentation Platform..... 19
- 3.11 Low-level Video Feature Extraction and indexing..... 20
 - 3.11.1 Description..... 20
 - 3.11.2 Rationale for Selection..... 20
 - 3.11.3 Integration in Documentation Platform..... 20
- 3.12 Tools diagrams..... 21
- 4 Software..... 24**
- 4.1 System Implementation..... 24
- 4.2 Software product packaging..... 24
 - 4.2.1 Package 1: JRS media-analyse suite..... 24
 - 4.2.2 Package 2: RAI Shotfinder..... 24
 - 4.2.3 Package 3: RAI Audio Analysis suite..... 24
 - 4.2.4 Package 4: RAI ARCHIE-CAS + MMStructure GAMP suite 24
 - 4.2.5 Package 5: University of Sheffield Editorial Segmenter.....25
- 4.3 System requirements and Installation..... 25
 - 4.3.1 Package 1..... 25
 - 4.3.2 Package 2..... 28
 - 4.3.3 Package 3..... 34
 - 4.3.4 Package 4..... 36
- 4.4 Launch an installation..... 38
 - 4.4.2 Package 5..... 47
- 5 Terms and conditions of use..... 49**
- 5.1.1 Package 1: Package 1: JRS media-analyse suite..... 49
- 5.1.2 Package 2: RAI Shotfinder..... 49
- 5.1.3 Package 3: RAI Audio Analysis suite..... 49
- 5.1.4 Package 4: RAI ARCHIE-CAS + MMStructure 49
- 5.1.5 Package 5: University of Sheffield Editorial Segmenter.....49
- 6 References..... 50**

1 Overview

1.1 Scope of the document

This deliverable is about the Content Analysis Tools delivered to the PrestoSpace Factory subsystem developed by the Metadata Access and Delivery Area. The tools described here are part of the MAD Documentation Unit [DEL],[DEL3.1].

This document is a part of a two-pieces deliverable, which includes in addition the software packages delivered as separate objects.

This document is made up of three main parts. The first part describes the tools selection criteria, and gives account of the major features and capabilities offered by the selected tools. This first part also gives the basic technical information about the software tools and illustrates the mappings between required functions and implementing tools.

The second part provides the sufficient information for the installation and run of the tools in the context of the PrestoSpace Documentation Unit. The last part covers legal aspects and licensing issues connected with the use of the software.

1.2 Selection Criteria

The following criteria have been used for decision about the inclusion of content analysis tools in the Documentation Unit:

- Relevance in the documentation process. Only tools that implement relevant functionalities with respect to the MAD Area Business Process Model [BPM], have been included after having been evaluated.
- Feasibility of the approach: Based on the findings in the state of the art report (D15.3, MDS3) only those tools have been selected, which are based on sufficiently mature technologies to be of practical use. The flexible plug-in concept of the documentation platform allows adding further tools in the future.
- Available know-how in the project: the selected tools are those for which there are partners in the PrestoSpace consortium that have sufficient know-how in the related fields of audiovisual content analysis. For other areas of audiovisual content analysis, where there are proven approaches, the flexible plug-in concept of the documentation platform may allow the integration of third party tools.
- User requirements: Users will not directly judge the usefulness of content analysis tools, but they will assess the automatically extracted annotations and supporting features (e.g. for visualisation and browsing) in the documentation and publication platforms. The results of the usability tests of these tools will provide feedback for the requirements on the content analysis tools.

2 Target Architecture

This part of the deliverable covers the description of the defined target logical architecture, enlightening and supporting the key design choices.

The central concept of the architecture consists in designing and developing an open and scalable structure for automatic content analysis, in order to allow flexible toolset

configuration management as well as platform and programming language independence. This goal is achieved through the definition and employment of a plug-in mechanism for the tools components and a service-based interface structure, e.g. for analysis workflow control and for shared content description, as defined in [ARCH].

Specifically, to support the requirements of the interfaces to manual annotation, semantic interpretation and the publication platform, the content analysis tools provide their data following the MAD Area data format, as specified in [MADDF]

3 Supported Tools

This part illustrates the set of specific content analysis tools included in the final realization of the Content Analysis subsystem of the PrestoSpace Documentation Unit. The content of this part substantially relies on the results coming out from the collection of the state-of-the-art of content analysis tools for video, audio and speech [MDS3], and from the relevance analysis conducted following the criteria explained in §1.2.

The following sections describe the tools individually:

- Shot boundary detection tools
- Key frame detection and extraction tools
- Stripe Images extraction tools
- Camera motion detection tools
- Visual activity extraction tools
- Speech to text transcription tools
- Audio structuring and segmentation tools
- Multimedia structure detection tools
- Editorial Parts segmentation tools
- Reference video clips detection tools
- Low-level visual features extraction tools

For each tools family, general features and functionalities are described at first, then a list of the software tools providing the described features is reported. Each software component is given a label, that will be used as reference in the software description sections, and benchmarking results are indicated where relevant and available.

3.1 Shot Boundary Detection

3.1.1 Description

Shot boundary detection is the MAD Area process function that produces a summarised view of the editorial object based on the visual shots.

Shot detection tools segment a video in its shots, which are its primary building blocks. They are capable of detecting both abrupt (cuts) and gradual transition (such as dissolves, fades,

wipes, etc.). The result produced is a decomposition of the timeline into shots (shot list). For gradual transitions, the start and end of the transition and its type can be determined.

3.1.2 Rationale for Selection

Shot boundary detection algorithms are sufficiently mature to be used in productive applications and know-how on shot boundary detection is available among the partners of the consortium. The applications of shot boundary detection results in the documentation platform are the following:

Prerequisite for other visual content analysis algorithms. Shot boundaries are natural limits of the occurrence of many mid and high level visual features, such as camera movements or object appearances. Thus the results of shot boundary detection will make these algorithms more robust and efficient.

Content structuring. As shots are the basic units of visual essence, the knowledge about shot boundaries can help to identify higher level structure elements like news stories or scenes. Shots can also serve as a unit of manual annotation.

Content indexing. The shot structure represents a fast and efficient indexing structure for accessing the content randomly.

Navigation support. The shot structure can be used to support a user in navigating through the video, both in manual documentation and viewing of query results.

3.1.3 Integration in Documentation Platform

3.1.3.1 Shot detection tool 1

3.1.3.1.1 Label

SD-MA.1

3.1.3.1.2 Brief technical description

The shot boundary detection implemented by this tool uses the essence and the basic technical metadata as input. The results are described using MPEG-7 in the timeline decompositions part of the editorial object document [MADDF].

3.1.3.1.3 Supported I/O formats

The I/O formats of the tools of the media-analyse GAMP are described in section 4.3.1.

3.1.3.1.4 Benchmarking

The shot boundary detection algorithm used by the tool has been evaluated on the TRECVID 2004 shot boundary detection extraction test set. The data consisted of about 65 hours of news material from 1998, containing 4,806 shot boundaries (57.7% cuts, 31.7% dissolves, 4.8% fade-out/fade-ins and 5.7% other transitions).

Table 3 shows the benchmarking results for our algorithm.

	Recall	precision
Cuts	0.776	0.693
gradual transitions	0.293	0.433
all transitions	0.620	0.596

Table 1: Shot boundary detection results.

3.1.3.2 Shot detection tool 2

3.1.3.2.1 Label

SD-SF.1

3.1.3.2.2 Brief Technical description

The shot boundary detection implemented by this tool uses the essence and the basic technical metadata as input. The results are described using MPEG-7 in the timeline decompositions part of the editorial object document [MADDF].

This tool is capable of detecting abrupt cuts and fast gradual transitions, shorter than about half-second, for which the given shot boundary is currently about the mid-point of the transition.

3.1.3.2.3 Supported I/O formats

The tool can be configured to use an external decoding chain (e.g. FFMPEG).

3.1.3.2.4 Benchmarking

Benchmarking has been performed for abrupt cuts on a set of RAI programmes. Results are shown in the following table.

	Recall	Precision
Cuts	0.93	0.93

Table 2: Shot boundary detection results.

3.2 Key Frame Detection

3.2.1 Description

The key frame extraction functionality delivers visualisation support to the shot detection functionality.

Key frame extraction tools extract a number of key frames from each shot, which are used as representative proxies for a shot's content. The key frame positions are selected either based on the visual activity in the visual essence, i.e. the more object and/or camera motion, the shorter the time between two key frame positions, or using simpler methods as taking the central and boundary frames of each shot. At least one key frame per shot is extracted. In order to make the key frames efficiently available for visualisation purposes, a downscaled version of each key frame is stored as an image.

3.2.2 Rationale for Selection

Key frames are an important means for efficiently representing the visual content of a shot and support navigation (e.g. for manual documentation, access and result presentation). The selection of the key frame position can be decided on the basis of a number of features from low- to high level. For reasons of both generality and efficiency, visual activity is a good compromise.

3.2.3 Integration in Documentation Platform

3.2.3.1 Key frame detection tool 1

3.2.3.1.1 Label

KF-MA.2

3.2.3.1.2 Brief technical description

This tool uses the essence and the results of the shot boundary detection tool SD-MA.1 as input. The outputs are the key frame positions and the visual feature descriptors, which are added to the MPEG-7 shot description of the editorial object document, and the key frame files, which are forwarded to the EMS via the standard Core Platform material insertion protocol.

Visual descriptors of the key frames are calculated using the sub-system of the documentation platform providing the low-level video feature extraction functionality (see § 3.10.3.1).

3.2.3.1.3 Supported I/O formats

The I/O formats of the tools of the media-analyse GAMP are described in section 4.3.1.

3.2.3.2 Key frame detection tool 2

3.2.3.2.1 Label

KF-SF.2

3.2.3.2.2 Brief technical description

This tool uses the essence and the results of the shot boundary detection tool SD-SF.1 as input. The outputs are the key frame positions and the key frame files, which are forwarded to the EMS via the standard Core Platform material insertion protocol. All the result data are integrated in the MPEG-7 part of the editorial object document.

This tool selects for each shot a picture taken from about the central position.

3.2.3.2.3 Supported I/O formats

The native format for key-frames images is JPEG. However, the tool can be configured to perform format conversion when forwarding the pictures to the EMS.

3.3 Stripe Image Extraction

3.3.1 Description

Stripe images are spatio-temporal representations of the visual content. The content of a fixed or moving column of the visual essence is visualized over time. This visualisation clearly shows shot boundaries as discontinuities and allows to quickly grasp the presence of camera or object motion.

3.3.2 Rationale for Selection

Stripe images can be efficiently extracted and serve as help for quick content overview and navigation, especially in the manual documentation tool but also in the result visualisation of the publication platform.

3.3.3 Integration in Documentation Platform

3.3.3.1 Stripe image extraction tool 1

3.3.3.1.1 Label

SI-MA.3

3.3.3.1.2 Brief technical description

This tool uses only the video essence as input. The outputs are a list of the extracted stripe image segments, which are added to the timeline decompositions of the editorial object document, and the stripe image files, which are forwarded to the EMS via the standard Core Platform material insertion protocol.

3.3.3.1.3 Supported I/O formats

The I/O formats of the tools of the media-analyse GAMP are described in section 4.3.1.

3.3.3.2 Stripe image extraction tool 2

3.3.3.2.1 Label

SI-SF.3

3.3.3.2.2 Brief technical description

This tool uses only the video essence as input. The outputs are a list of the extracted stripe image segments, which are added to the timeline decompositions of the editorial object document, and the stripe image files, which are forwarded to the EMS via the standard Core Platform material insertion protocol.

3.3.3.2.3 Supported I/O formats

The native format for stripe-images is JPEG. However, the tool can be configured to perform format conversion when forwarding the pictures to the EMS.

3.4 Camera Motion Detection

3.4.1 Description

The goal of this functionality is the detection of segments containing salient camera motion information and to describe this motion both qualitatively and quantitatively, as well as the boundaries of these segments. The original camera motion cannot be fully recovered from the analysis of the result video, as for instance it is not possible to discriminate a translation of the camera (e.g. a track left) from a rotation (e.g. a pan left), if the scenery is close to the camera. The result is thus the described using four types of camera motion (pan, tilt, zoom, roll) and a rough quantisation of the amount of motion.

3.4.2 Rationale for Selection

The detection of camera motion is useful for the following purposes:

Camera motion can be used to infer higher level information, if combined with other analysis results or domain knowledge. For example, zooming on an object or person is an indicator of relevance, and in field sports, pans indicate the direction of the game.

Search criteria for re-use of material. Visual grammar imposes constraints on the camera motion of sequences to be combined. When searching for essence in order to re-use it in new productions, camera motion is an important search and selection criteria.

Navigation help. Camera motion annotation can guide the user and support manual annotation.

The analysis of the state of the art of camera motion has shown, that – at least on news material – the approaches are reliable enough to be used in productive applications.

3.4.3 Integration in Documentation Platform

3.4.3.1 Camera Motion detection tool 1

3.4.3.1.1 Label

CM-MA.4

3.4.3.1.2 Brief technical description

This tool uses the essence and the results of shot boundary detection provided by the tool SD-MA.1. The output is the description of the camera motion segments per shot (added to the MPEG-7 shot decomposition) and the salient types of camera motion in each segment and their respective strengths.

3.4.3.1.3 Supported I/O formats

The I/O formats of the tools of the media-analyse GAMP are described in section 4.3.1.

3.4.3.1.4 Benchmarking

The camera motion detection algorithm used by the CM-MA.4 tool has been evaluated on the TRECVID 2005 low-level feature extraction test set. The data consisted of 2004 news material, of which about 5000 shots have been considered by NIST as being used as ground truth. 2226 of these shots have been found to contain clearly determinable motion by the

human annotators. Of those 1159 shots contained no motion, and 587 pans, 210 tilts and 511 zooms have been annotated.

Table 3 shows the benchmarking results for the two submitted runs using our algorithm.

	Pan		Tilt		Zoom		Mean ¹	
	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
Run 1	0.927	0.606	0.831	0.657	0.610	0.947	0.789	0.737
Run 2	0.919	0.676	0.828	0.686	0.640	0.933	0.796	0.765

Table 3: Results of the submitted runs.

3.5 Visual Activity

3.5.1 Description

Visual activity is a measure for the change in the visual essence due to global or local motion. The result of the is a measure of the visual change over time.

3.5.2 Rationale for Selection

The visual activity measures can be used as a hint for scenes containing action. Using the results of camera motion detection, it is possible to identify scenes containing object motion. This can both be used by subsequent analysis modules and to support the user in manual documentation.

3.5.3 Integration in Documentation Platform

3.5.3.1 Visual activity tool 1

3.5.3.1.1 Label

VA-MA.5

3.5.3.1.2 Brief technical description

This visual activity tool is integrated in the documentation platform by. The tool uses the essence and the results of shot boundary detection provided by SD-MA.1. The output is the description of the visual activity over time (added to the MPEG-7 shot decomposition).

3.5.3.1.3 Supported I/O formats

The I/O formats of the tools of the media-analyse GAMP are described in section 4.3.1.

¹ The calculation of the mean precision and recall in this evaluation does not consider the different frequency of the types of camera motion, it is simply calculated as the average of the results for each type.

3.6 Speech to Text Transcription

3.6.1 Description

Speech to text transcription is the process function that produces a text transcript of spoken language present in the audio channels of the essence. As a preparatory phase Audio Structuring and Segmentation [3.7] is performed, so that only segments categorised as spoken parts are considered. In literature the whole process is also known as ASR, Automatic Speech Recognition.

Supported languages are:

- Italian
- English

For the Italian language is possible to manage and customize a dynamic vocabulary (max 4K words).

The quality of the produced output is affected by the lacking of punctuation and capitalisation and by a certain number of missing or erroneous words depending on :

- Algorithms
- Signal quality
- Phonetic model
- Language model (65K words)

All these aspects should be taken into account if the output data are used for further elaborations.

3.6.2 Rationale for Selection

The availability in form of text of the spoken content is relevant for several reasons:

Text-based content search and retrieval. The result of speech to text transcription is directly available for traditional full-text searches and retrieval.

Support to higher level information extraction. Transcribed text is the primary source from which to infer, through various natural language processing techniques, other content-related information, e.g. content classification, named entities detection.

Content indexing. If transcribed text is made available on the publication platform, during content navigation, word-level and sentence level synchronisation information can be used to randomly access the content based on the users' interests.

3.6.3 Integration in Documentation Platform

3.6.3.1 Speech to text tool 1

3.6.3.1.1 Label

STT-ITC.1

3.6.3.1.2 Brief technical description

This tool uses the audio content as the source for the detection and transcription of text, relying on the audio segment classification services provided by ASS-ITC.2,

The component has been integrated in the platform via GAMP interface and the result put in the EDOB file as MPEG7 timeline decomposition layers for audio derived text content.

3.6.3.1.3 Supported I/O formats

It takes as input an uncompressed PCM audio file (internally converted into sphere 16 bit 16 Khz mono).

Outputs include the transcript text, which is placed into the EDOB format as an MPEG7 timeline decomposition layer (time synchronisation at word level and sentence level are supplied).

3.6.3.1.4 Benchmarking

Evaluations have been made on a training set of newscast programmes. Results indicate a precision of 85% on studio parts (news announcers) and 70% on external contributions (in the latter case performance falls down because of environmental noise, superimposed speakers, not planned speech).

	Precision	Recall
Studio	0.85	N/A
External contribution	0.70	N/A

Real time performances (1 hour processing time for one hour of input material) can be achieved with a cluster of two hosts based on dual processor dual core architecture with 2GB of RAM. The audio content has to be provided to the processor as a complete and closed file.

The algorithm maps a sequence of phonemes against words stored in the vocabulary so that in the output cannot appear extraneous words. The major part of elaboration time is due to the exploration of the bi-grams and tri-grams trees representing the language model (conditional probabilities for sequences of 2 or 3 words).

The language model is domain specific (e.g. newscast domain) and has to be derived in an appropriate training session.

3.7 Audio Structuring and Segmentation

3.7.1 Description

Audio structuring and segmentation is a process function that produces a segmentation of the multimedia content based on the acoustical classification.

Segments can be classified as:

- music;
- speech;
- silence;
- noise;

For speech sections it is also supplied a segmentation based on:

- gender classification (male/female speech)
- acoustical quality (wideband/narrowband signal)
- speaker labelling
- sentences (based on speech breaks)

During speaker labelling, speakers identified with sufficiently similarity are given the same label all throughout the timeline.

3.7.2 Rationale for Selection

Speaker labelling is a useful feature that can be used as input information for higher level information extraction tasks (as multimodal content clustering) as well as in search and retrieval tasks. Audio segmentation based on acoustical classification can be useful both for content indexing and for search and retrieval as an additional selection criterion in the formulation of queries.

3.7.3 Integration in Documentation Platform

This module is part of Speech to Text Transcription suite [3.6].

3.7.3.1 Audio structuring and segmentation tool 1

3.7.3.1.1 Label

ASS-ITC.2

3.7.3.1.2 Brief technical description

The tool performing this function saves results of audio structuring and segmentation using a single temporal decomposition layer of the MAD Data Format. Native labels used by the software are preserved.

3.7.3.1.3 Supported I/O formats

It takes as input an uncompressed PCM audio file (internally converted into sphere 16 bit 16 KHz mono).

Outputs include the audio structure segmentation timeline mapped into EDOB format as an MPEG7 decomposition layer.

3.7.3.1.4 Benchmarking

No reliable scores for recall/precision are available at the moment. Sometimes speaker over segmentation has been noticed.

The task execution is very light (two order of magnitude lower if compared to Speech to Text). As an example 1 hour of input material can be processed in 3 minutes on a dual processor server.

3.8 Multimedia Structure Detection

3.8.1 Description

Multimedia structure detection is a function whose aim is to discover recurring multimedia (i.e. aural, visual or audiovisual) structures in an audiovisual content. This means that events like similar shots or the same speaker talking are detected and that the corresponding temporal boundaries are taken as discontinuity markers, thus allowing for a timeline decomposition of the material.

Multimedia structure detection is a specific task of the more general content structure detection, aimed at discovering recurring data configurations drawn from the various media channels (audio, video, spoken text), in order to construct elementary or compound indexing structures to the content.

3.8.2 Rationale for Selection

The multimedia structure detection functionality is used, in PrestoSpace MAD, as an input information to the Editorial Part Segmentation task (§3.9). In fact, in programmes where the audiovisual structure is an important layer to explain part of the editorial structure (e.g. newscasts), multimedia structure provides a very good base of information. Multimedia structure can alternatively be used as an editorial segmentation functionality per se, as done by EP-CAS.1.

3.8.3 Integration in the Documentation Platform

3.8.3.1 Multimedia Structure Detection tool 1

3.8.3.1.1 Label

MS-CAS.4

3.8.3.1.2 Brief Technical Description

This tool detects multimedia structure elaborating the audio and video channels of a material in a hierarchical fashion. At first it detects visual shots clusters and performs a filtering on them, based on some of their structural features (e.g. elongation, duration, total number of shots). Then it applies a second filter on the reduced shot clusters set by selecting those clusters which have a defined temporal coverage w.r.t. the main speakers. Main speakers are those whose temporal coverage of the programme's timeline exceeds a determined threshold, and are identified using the outputs provided by ASS-ITC.2. This cascade two-filter method is tailored on the subsequent editorial segmentation task, and aims at (heuristically) discarding the part of the total available multimedia structure with less probability of being meaningful from the editorial point of view.

3.8.3.1.3 Supported I/O formats

The output format is a temporal decomposition of the EDOB document, conforming to the PrestoSpace MAD data format specification. Input formats are those compatible with the used audiovisual decoding chain (FFMPEG). Refer to specific documentation to have a complete list of accepted formats (<http://ffmpeg.mplayerhq.hu/>). Outputs of the Media structure detection tool are integrated in the EDOB document being elaborated using a Temporal Decomposition of the root AudioVisual element with criteria "multimedia structure".

3.8.3.1.4 Benchmarking

Benchmarking results for video structure detection are available on a set of RAI broadcast news (approx. 10 hours of material).

Precision	Recall	F-Measure
0.79	0.97	0.86

Table 4. Multimedia structure tool benchmarking.

3.9 Editorial Parts Segmentation

3.9.1 Description

Editorial parts segmentation is the task of identifying the temporal segments of an editorial object that are its constituent parts from the editorial perspective.

3.9.2 Rationale for Selection

Segmentation of an editorial object content in its editorial constituent parts is a core feature that any documentation system should have among its baseline functionalities. The definition of what constitutes an editorial part depends on several factors, among which the category of the programme (e.g. news, sports, entertainment) plays the most important role. Scientific and technical literature concerned with automatic editorial segmentation provide only partially satisfactory results, relegating higher figures only to few well-established categories (e.g. news programmes) [MDS3]. Even if fully automatic segmentation may not be possible, these techniques can be used as aiding means to the human annotators. This is the principal reason for which such tools are considered in the PrestoSpace MAD documentation platform.

3.9.3 Integration in Documentation Platform

3.9.3.1 Editorial parts segmentation tool 1

3.9.3.1.1 Label

EP-CAS.1

3.9.3.1.2 Brief technical description

This tool performs editorial parts identification based on video features analysis and speaker labeling only, in the restricted domain of extended format newscast programmes (e.g. the main newscast evening edition of the general public broadcast channels). Low level features information are provided by LLV-CAS.3 The tool uses the speaker labelling information produced by ASS-ITC.2 to improve the precision of the results. This tool is a particular application of MS-CAS.4, e.g. multimedia structure is assumed to be coincident with editorial segmentation.

3.9.3.1.3 Supported I/O formats

The output format is a temporal decomposition of the EDOB document, conforming to the PrestoSpace MAD data format specification. Input formats are those compatible with the used audiovisual decoding chain (FFMPEG). Refer to specific documentation to have a complete list of accepted formats. Outputs of the tool are integrated in the EDOB document

being elaborated using a Temporal Decomposition of the root AudioVisual element with criteria "Editorial Parts".

3.9.3.1.4 Benchmarking

The tool has been made run on a set of programmes used as a test set for PrestoSpace Documentation Platform tests and on a set of RAI newscast programmes. Results are summarised in the following table. RAI programmes were selected among RAI's main newscast editions, while the PrestoSpace test set was integrated with a set of BBC's news bulletins.

	Precision	Recall
RAI programmes	0.64	0.64

Table 5. RAI editorial segmenter benchmarking results.

In the RAI test case, the tool reaches almost state of the art (cfr. MDS3, section 14.2.1).

3.9.3.2 Editorial parts segmentation tool 2

This tool is intended to combine a set of pre-existing segmentations of an editorial object into a single unified authoritative one. The rationale behind it is that a set of tools can be used to infer different segmentations based on different types of cues, e.g. by analysing the lexical cohesion of the ASR transcript, by looking at speaker changes in the ASR stream, by clustering shots based on visual features, etc. Once all these candidate segmentations have been produced, this tool can then be used to create a unified segmentation that exploits all the information available with the aim of improving the quality of the final segmentation.

3.9.3.2.1 Label

EP-SHEF.1

3.9.3.2.2 Brief technical description

This tool implements a weighted voting algorithm for combining an arbitrarily large set of input segmentations into a single output segmentation. Each input segmentation has an associated weight value which is combined with the confidence value of each individual split point contained in that segmentation. The tool configuration comprises two main parameters:

- a **window** value: which decides the level of *fuzziness* that the tool allows (what is the maximum distance between two split points that are still considered compatible).
- a **threshold** value: which defines the minimum value for the weighted sum of confidence values for an output split point to be inferred.

The algorithm is illustrated in Figure 3.1. Starting from a set of candidate segmentations (represented as timelines with split points), an imaginary sweeping window (with the width specified by the **window** parameter) is moved from the start to the end of the original editorial object timeline. At each point, a confidence value is calculated using the following formula, for all the split points included in the window where w_i is the weight associated with the segmentation i , and c_{ij} is the confidence value for the split point j from segmentation i :

$$\sum w_i * c_{ij}$$

Whenever this sum surpasses the specified **threshold** value, an output split point is generated.

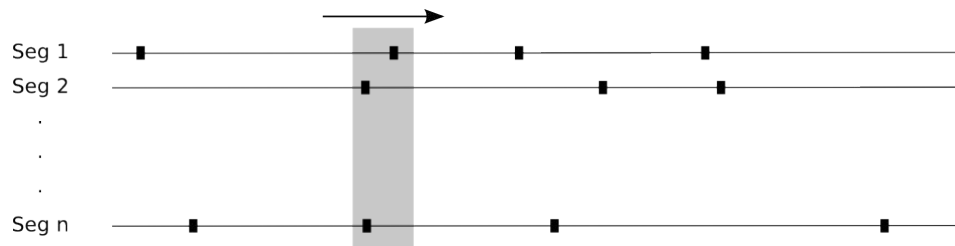


Figure 3.1. Segmentation Algorithm

The values for these two parameters, together with the weights used for the input segmentations, affect the way the tool uses the input split points to generate the output segmentation. Here are two corner cases to illustrate the changes in the tool behaviour based on the configuration provided:

1. When all input weights are 1 and threshold value is n , then an output split point is only generated when **all** input segmentations “agree”, i.e. they all provide split points inside a window. This would generate the output segmentation with the least number of split points.

When all input weights are 1 and threshold is also 1, then an output split point is generated for each input split point from each input segmentation. Note, however, that all input split points that are contained inside one window will generate a single output split point, in other words, the shortest possible output segment length is equal to the value of the **window** parameter. This would generate the output segmentation with the largest number of split points.

3.9.3.2.3 Supported I/O formats

The tool uses temporal decompositions in the EDOB document, as defined by the MAD data format specification, for both its input and output.

3.9.3.2.4 Benchmarking

This tool simply combines the output of several other tools that are run previously. Its behaviour is fully deterministic and depends wholly on the values provided as input, so it has no intrinsic function that is suitable for evaluation.

3.10 Reference Video clip detection

3.10.1 Description

Reference Video clip detection is the task consisting in identifying occurrences of a reference video sequence in another independent video sequence.

3.10.2 Rationale for Selection

Reference Video clip detection, although a simple task, can yield very important results in processes of material acquisition and restoration (e.g. by automatically identifying colour bars, noise) and in support of the documentation process in cases of programmes containing particular sequences (e.g. jingles, headers, graphics, commercials) used as separators/indicators of certain programme events.

3.10.3 Integration in Documentation Platform

3.10.3.1 Reference video clip detection tool 1

3.10.3.1.1 Label

RVDD-CAS.2

3.10.3.1.2 Brief technical description

This tool uses an updateable library of video segments features set used as references for detecting and identifying matching instances in input video sequences. Results can be transformed to the MPEG-7 temporal decomposition syntax adopted by the MAD Data Format. It uses the feature extraction services provided by LLV-CAS.3. At the current time for this product, no GAMP integration has been made with the documentation platform, although the software itself is being delivered to the project. Future developments may include this piece of integration.

3.10.3.1.3 Supported I/O formats

The output format is a temporal decomposition of the EDOB document, conforming to the PrestoSpace MAD data format specification. Input formats are those compatible with the used audiovisual decoding chain (FFMPEG). Refer to specific documentation to have a complete list of accepted formats.

3.10.3.1.4 Benchmarking

A test platform for video clip detection based on the RVDD-CAS.2 tool has been set up in RAI for evaluating the performances in some relevant cases. These cases include: detection of black video segments, detection of advertisements, detection of newscast programmes jingles. Table 6 summarises the results in terms of precision and recall in the case of advertisements spots detection.

	Precision	Recall
RAI advertisement spots detection	0.79	0.56

Table 6. Ad spot detection benchmarking results.

Based on the same tool, a segmentation system has been set up and tested. This system had the objective of a) detecting advertisements insertions in channels' broadcast schedules b) segmenting individual spots in the advertisements insertions. Results are summarised in Table 7.

	Precision	Recall
RAI advertisement insertions segmentation	0.78	0.87

Table 7. Ad insertions segmentation benchmarking results.

3.11 Low-level Video Feature Extraction and indexing

3.11.1 Description

Low-level video feature extraction is the process function responsible of providing descriptions of the video component of material by means of standardised feature descriptors expressed in MPEG-7. Features include colour, texture and motion descriptors.

Visual descriptors of the colour and texture properties of the key frames or another subset of frames can be extracted as a compact metadata representation of the visual content of a shot. The descriptors may be used to determine visual similarity between shots. Shot-level descriptors can be generated from the frame level results (e.g. average colour histogram).

3.11.2 Rationale for Selection

Low-level video feature extraction is basically useful for the extraction of higher level metadata or for the implementation of retrieval systems based on the query by example paradigm. The description of the visual properties of frames and/or shots using the standardized MPEG-7 visual descriptors allows to efficiently determine similarity in terms of global visual properties between frames or shots from different parts of a video. The visual similarity between shots serves as a hint for similar content and similar settings and thus supports content structuring and navigation, in addition to serving as an input to higher level elaboration.

3.11.3 Integration in Documentation Platform

3.11.3.1 Low-level video feature extraction tool 1

3.11.3.1.1 Label

LLV-MA.6

3.11.3.1.2 Brief technical description

This tool produces various colour features descriptors associated to the key frames extracted by KF-MA.2. These features may be used as indexing data for retrieval applications based on video segments similarity.

3.11.3.1.3 Supported I/O formats

The I/O formats of the tools of the media-analyse GAMP are described in section 4.3.1.

3.11.3.2 Low-level video feature extraction tool 2

3.11.3.2.1 Label

LLV-CAS.3

3.11.3.2 Brief technical description

This tool provides various colour and texture features as well as temporal information (displaced frame differences). It is also capable of providing features descriptors for arbitrary segments, i.e. not limitedly to the shot detection boundaries. These features may be used as indexing data for retrieval applications based on video segments similarity. All features are provided on a global frame basis histogram, or on an image grid decomposition basis (block dimensions=16x16).

3.11.3.2.3 Supported I/O formats

Input formats are those compatible with the used audiovisual decoding chain (FFMPEG). Refer to specific documentation to have a complete list of accepted formats. Output Data are currently not integrated in the EDOB document.

3.12 Tools diagrams

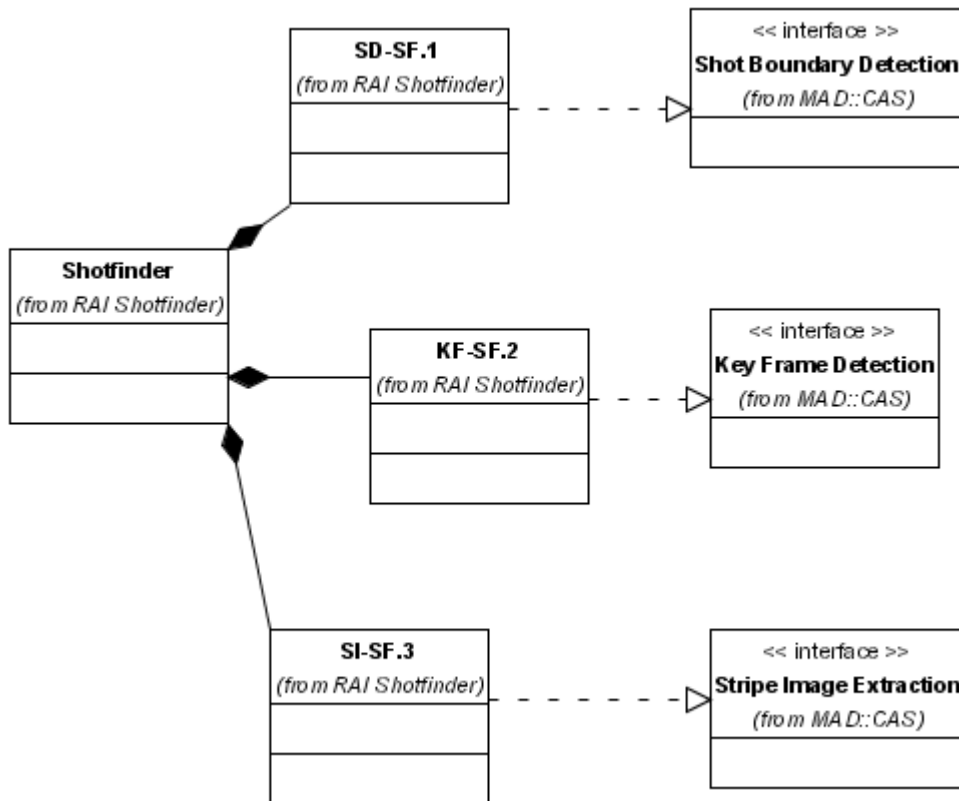


Figure 2. RAI shotfinder package structure

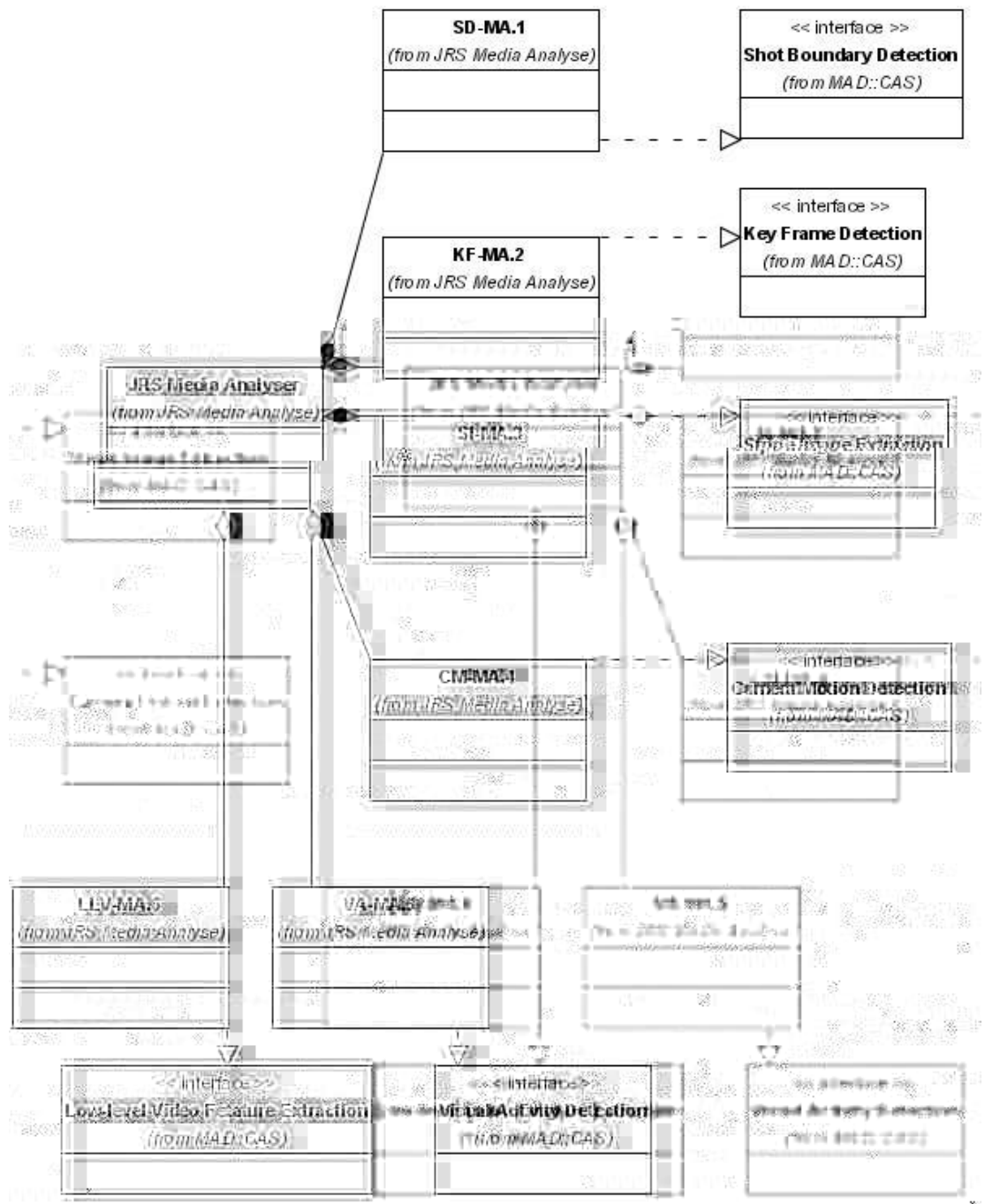


Figure 3. JRS media Analyse package structure

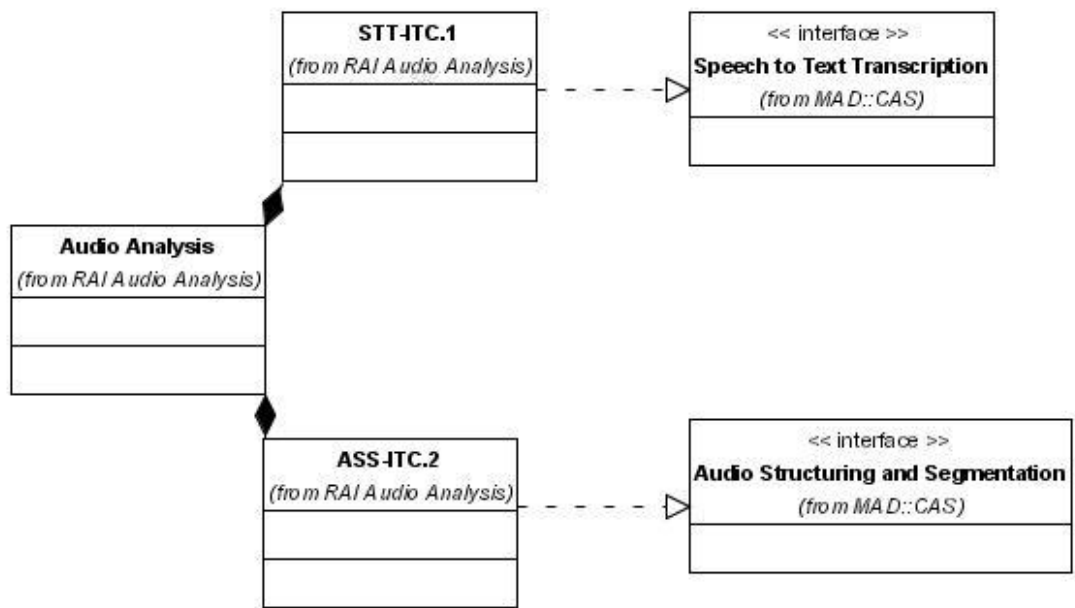


Figure 4. RAI Audio Analysis package structure

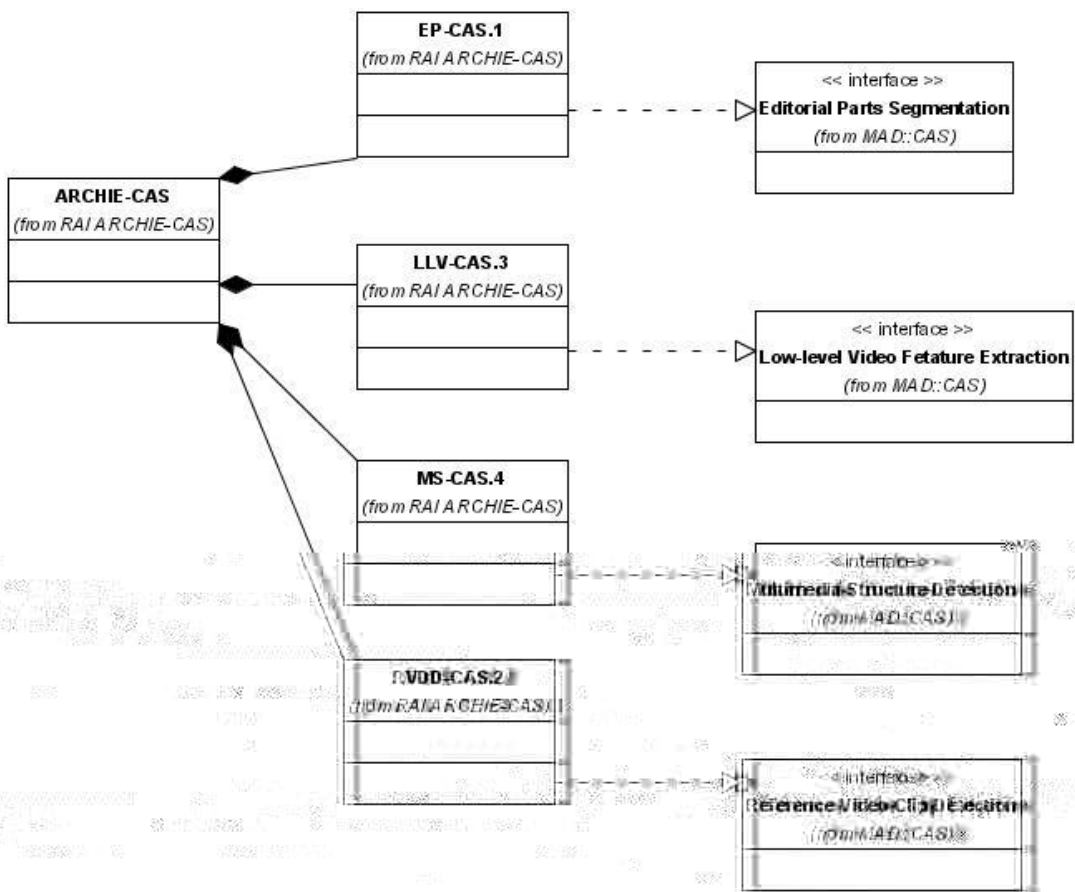


Figure 5. RAI ARCHIE-CAS package structure

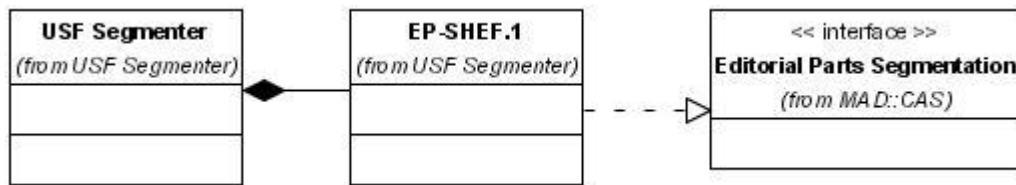


Figure 6. USF editorial segmenter package structure

4 Software

This part delivers the full technical documentation of the software. This includes software packaging documentation, system requirements specifications, and installation instructions for each of the packages.

4.1 System Implementation

4.2 Software product packaging

4.2.1 Package 1: JRS media-analyse suite

This package includes the following tools: **SD-MA.1**, **KF-MA.2**, **SI-MA.3**, **CM-MA.4**, **VA-MA.5**, **LLV-MA.6**.

In addition, the package contains a document server component that collects the MPEG-7 snippets produced by the analysis tools in the package, and serves as a metadata interface to the EMS.

4.2.2 Package 2: RAI Shotfinder

This package includes the software components implementing the following tools: **KF-SF.2**, **SD-SF.1**, and **SI-SF.3**

4.2.3 Package 3: RAI Audio Analysis suite

This package includes the software components implementing the following tools: **STT-ITC.1** and **ASS-ITC.2**.

4.2.4 Package 4: RAI ARCHIE-CAS + MMStructure GAMP suite

This package includes the software components implementing the following tools: **EP-CAS.1**, **LLV-CAS.3**, **MS-CAS.4**, **RVDD-CAS.2**.

4.2.5 Package 5: University of Sheffield Editorial Segmenter

This package includes the **EP-SHEF.1** tool as a standalone Java application implementing a GAMP that communicates with the web services of the MAD documentation platform via SOAP messages.

4.3 System requirements and Installation

4.3.1 Package 1

4.3.1.1 Operating system requirements

The media-analyse GAMP runs on Windows 2000, Windows 2003 Server or Windows XP.

4.3.1.2 Environment software requirements

DirectX 9 runtime or newer must be installed.

All other third party components that are required by the media-analyse GAMP are packaged with the software distribution.

4.3.1.3 User guide

4.3.1.3.1 Installation

The software package comes with an installation tool that copies the file to the selected locations on disk.

The document server is installed as a Windows service, which is by default started automatically (the settings can be changed in the Windows services configuration).

If the Windows firewall is activated, make sure to define exception to allow communication between the GAMP and the document server as well as between the GAMP/document server and the MAD core platform.

4.3.1.3.2 Configuration

A simple configuration tool is installed with the package. Please stop all applications of the package before making configuration changes. The configuration tool must be run before the first use of the GAMP. Table 8 shows the settings that can be modified with the configuration tool.

Parameter	Description
Document Server Host	Host name or IP address of the machine on which the document server is installed (typically local machine, i.e. IP 127.0.0.1).
Document Server Port	Port that is used for the document service.
EMS Service URL	URL of the EMS web service.
Workflow Service URL	URL of the workflow web service.
Workflow queue name	Name of the queue of the workflow service that is

	queried for jobs.
Workflow service polling interval	Interval (in milliseconds) for polling the workflow service for new jobs.
Service user name	User name of the GAMP for authorization at the MAD core platform services.
Service password	Password of the GAMP for authorization at the MAD core platform services.
FTP host	Hostname/IP address of the FTP server to which produced materials are copied for making them available to the core platform for check-in.
FTP username	Username at the FTP server to which produced materials are copied for making them available to the core platform for check-in.
FTP password	Password at the FTP server to which produced materials are copied for making them available to the core platform for check-in.
Local data mount point	Local root directory of the data area (paths are specified relatively to this location).
Local meta-essence mount point	Local root directory of the data area for produced key frames and stripe images (paths are specified relatively to this location).
EMS data mount point	Root directory of the data area as seen from the EMS (e.g. FTP URL) (paths are specified relatively to this location).
EMS meta-essence mount point	Root directory of the data area for produced key frames and stripe images as seen from the EMS (e.g. FTP URL) (paths are specified relatively to this location).

Table 8: Configuration of the media-analyze installation.

4.3.1.3.3 I/O formats

Video input formats: MPEG-1, MPEG-2, DV, Windows Media 9. Note: for MPEG elementary streams, the progress report may be incorrect, as presentational time stamps are not available.

Audio input formats: WAV PCM, MP3, BWF

Essence output format: Key frames and stripe images are stored as JPEG files.

Metadata output format: The metadata output is inserted into the EDOB format according to the MAD data format specification.

4.3.1.3.4 Usage

The media-analyse GAMP contains a set of modules that have interdependencies. The module graph is shown in Figure 7. As listed in the parameter table below, modules can be deactivated, but this may also result in deactivation of dependent modules.

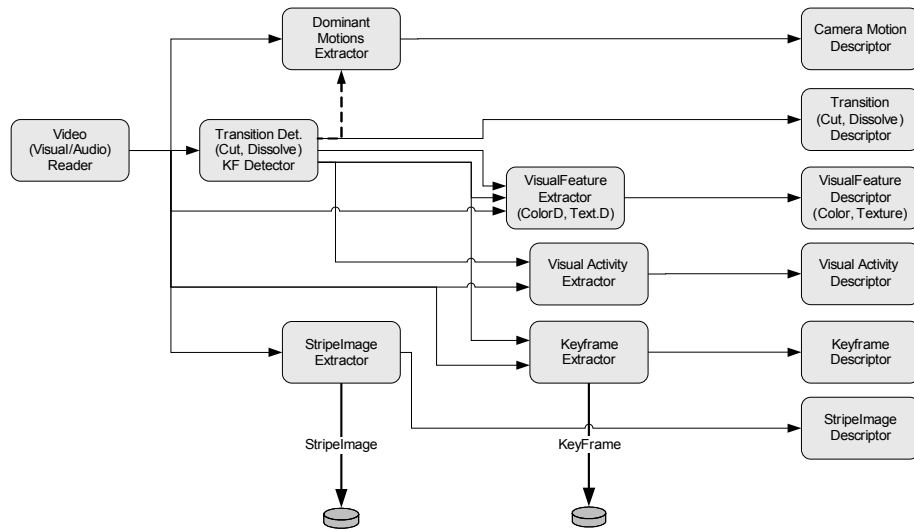


Figure 7: Audiovisual processing graph currently implemented in the media-analyze GAMP.

The media-analyze GAMP has a simple user interface that shows the items that have been processed or are currently being processed. A log window shows the output of the processing modules for the current item.

A number of configuration options can be set for the GAMP (passed via the job XML). The options are listed in Table 9.

Name	T ²	possible values ³	Description	Dependencies
KeyFrames	P	<u>on</u> , off	perform key frame detection (and thus feature extraction from key frames)	
FeatureExtraction	P	<u>on</u> , off	perform visual feature extraction from key frames	requires KeyFrames=on
GradualTransitions	P	<u>on</u> , off	detect both hard cuts and gradual transitions	
StripImages	P	<u>on</u> , off	perform stripe image extraction	
CameraMotion	P	<u>on</u> , off	perform camera motion detection	
MotionActivity	P	<u>on</u> , off	measure motion activity	
SBDHigherRecall	O		optimize shot boundary detection for recall	opposite of SBDHigherPrecision
SBDHigherPrecision	O		optimize shot boundary	opposite of

² Type: **Option** (i.e. switch) or **Parameter** with value.

³ The underlined value is the default. Options are by default not set.

			detection for precision	SBDHigherRecall
KeyframeDensity	P	High, <u>medium</u> , low	Temporal density of key frames inside shot	requires KeyFrames=on
SmallerKeyFrames	O		reduce size of key frames to 1/8 CIF (default is QCIF)	requires KeyFrames=on
SmallerStripeImages	O		Reduce size of stripe images to 256x64 (default is 512x128)	requires StripeImages=on
CMHigherRecall	O		optimize camera motion detection for recall	opposite of CMHigherPrecision
CMHigherPrecision	O		optimize camera motion detection for precision	opposite of CMHigherRecall

Table 9: Job configuration options of media-analyse GAMP.

4.3.2 Package 2

4.3.2.1 Operating system requirements

The RAI ShotFinder suite (GAMP CA_shots) suite run entirely on GNU/Linux platform.

Developments, tests and implementations used in PrestoSpace have been deployed and run on RedHat 8.0/RedHat 9.0 and Fedora Core 3 distributions.

Although there are not specific reasons that prevent the use on other Linux distributions or Unix OS, RAI does not ensure reliability of installations made on platform different from those tested.

4.3.2.2 Environment software requirements

4.3.2.2.1 Core Process requirements

The core processes use:

- ffmpeg - for decoding the video material
- libjpeg -
- ImageMagick (convert command) - for still picture format and size conversions

4.3.2.2.2 Gamp architecture requirements

The Gamp uses:

- The GenericGamp utility in order to communicate with the Documentation Core Platform
 - This utility requires a Java Run Time Environment according to its version.
- linux command xsltproc for XML transformations, which is usually provided together with the Linux distribution. Libraries libxml, libxslt, and libexslt.

- ruby interpreted language for executinng scripts. This is usually provided together with the Linux distribution.
- A mechanism for accessing the video material sources made available by the EMS. This may change according to the CorePlatform configuration and on the overall deployment architecture of the Documentation Units. A fully equiped system should include:
 - ability to access files on http, https, or ftp protocols
 - wget command
 - abiltity to access files on Shared file system in one of these two ways
 - mounted as smbfs or cifs
 - smbclient
 - ability to access files on NFS file system
 - ability to access files on other systems through ssh
- A mechanism to make available to the EMS the key-frames and the stripe-images produced. This may change according to the CorePlatform configuration and on the overall deployment architecture of the Documentation Units. Unless the CorePlatform and the Gamp both run on the same system there should be at least one among:
 - ssh granted access from EMS, with write permission, and use of scp linux command. This approach was uded during the PrestoSpace tests.
 - granted access from EMS, with write permission, through other protocols (http, ftp, smb). It requires the specific client
 - grant ssh access to the EMS

grant access to the EMS through another protocol (http, ftp, smb). It requires the specific server

4.3.2.3 User guide

4.3.2.3.1 *Installation and setup*

The package is available in tar.gz format and it includes all the scripts, the xsl stylesheets, and the tools for the shot detections and for stripe-images extractions.

Those tools were developed in C language and the package may include them either either in binary form, pre-compiled for a target distribution, or with source code.

4.3.2.3.1.1 *installation steps*

- log as root user – (this is necessary only for the last part of installation)
- unpack the tarball in a folder of your choice. For instance */usr/local*, henceforth referenced as *\$(yourfolder)*
- move to the newly created directory “*GampShots*” and run
 - make - check for errors
 - make install
- The executable will be linked into */usr/local/bin* folder. You can change the settings by editing the makefile
- make sure that the folder */usr/local/bin* (or what you set) is in the path

4.3.2.3.1.2 *setting the environment*

Most environment variables can be set by editing with a text editor the file `$(yourfolder)/GampShots/etc/shots.conf`, henceforth referenced as *shots.conf*

4.3.2.3.1.2.1 GenericGamp

- Into *shots.conf*, set the variable *GenericGampBinFolder* to the executable path of GenericGamp previous installed. Example:

`/usr/local/GenericGAMP_linux_2.1/bin`

4.3.2.3.1.2.2 Connection to CorePlatform

open with a text editor the file `$(yourfolder)/GampShots/etc/mad_connection.xml` and modify according to your CorePlatform indications.

- URL of CorePlatform WorkFlow web service
- URL of EMS web service
- User / password for accessing

4.3.2.3.1.2.3 Running folder

You have to indicate the folder where the gamp will write output data on your local system.

- Into *shots.conf*, set the variable *ShotsCoreRunningFolder* to the desired value

4.3.2.3.1.2.4 Picture delivery to EMS

Current version copies the jpeg files to the EMS with scp command and then the InsertMaterials tells where files are on the EMS system (locally to EMS).

- Into *shots.conf*, set properly the values variables *remoteserver*, *remoteuser*, and *remotedir* within the *Copy2Server* section
- you'll also have to generate and load to the EMS system the ssh public key of the user that will run the Gamp

4.3.2.4 Running the Gamp

4.3.2.4.1 *User & permissions*

The Gamp can be made run by any system user provided that he/she has write permission on the Running folder, and he/she is granted for Picture delivery to EMS.

However it is recommended that the Gamp is not run by user *root*.

4.3.2.4.2 *Launching the Gamp*

The syntax for launching the Gamp is simply :

`shots.gamp <queue_name>`

where *<queue_name>* is the name of the CorePlatform Work-Flow queue that will be polled for jobs by the Gamp.

The standard output of the process includes a log of the operations at the top Gamp level and it may be useful to redirect it to a log file.

The Gamp can be launched by any position of the file system, provided that the user has write permission. However the main output data will be produced into the Running folder.

4.3.2.4.3 Stopping the Gamp

The Gamp normally runs an infinite loop.

In case you want to stop the Gamp, you simply have to look at the output and kill it when it is idle.

To kill the Gamp when it is busy can put the Working Units within the CorePlatform into a locked state.

4.3.2.4.4 Working Exceptions

Normal exceptions during the Gamp operation are managed and appropriate failure messages are notified to the CorePlatform.

Exceptions are also reported on the standard output.

4.3.2.5 Working options & working profiles configurations

The Gamp can receive working instructions from the CorePlatform Work-flow according to Table 10.

Option/Parameter	Behaviour
HigherPrecision	Gamp will use an HigherPrecision profile setting for shot detection. Default is BestCompromise
HigherRecall	Gamp will use an HigherRecall profile setting for shot detection. Default is BestCompromise
KeyFrames=off	Unsupported option Keyframes are not delivered to EMS.
SmallPictures	Pictures delivered have size reduced by 50%x50%
StripImages=on	The extraction of stripe images is activated. Default is off

Table 10 - gamp shots working options

The command line options for the specific tools are saved in *shots.conf* and can be modified for low level control.

For the Shot Boundaries detection three different settings are identified:

- BestCompromise – Default setting
- HigherRecall – more shots but less sure they are right
- HigherPrecision – less shots but more sure they are right

4.3.2.5.1 Shotfinder command line options

Command line options	Meaning	Suggestion
-f gray grayp 422 422p 420p	Tells which is the format of video input. It must match with decoder output.	- f 420p
-s <size>	Tells geometry of video input, on Field Base. It must match with decoder output	-s 720x288 for standard resolution
-m (run mode) play afap	Run mode can be play (if input is real time from hardware) or as fast as possible (afap, when input is from a video file decoding)	-m afap
-j quality <75> smooth <0>	jpeg quality configuration	Keep default
-t on off area <8x8> nodetect <11> soglia <2.0> moffset <2.5>	hard cut configuration : <ul style="list-style-type: none"> ○ nodetect – the detection is blinded for a few frames after a cut ○ soglia – greater value for higher precision ○ moffset – greater value for higher precision 	See defined profiles

<p>-d on off area <8x8> nodetect <30> soglia <2.0> moffset <0.5></p>	<p>Fast dissolve configuration :</p> <ul style="list-style-type: none"> o nodetect – the detection is blinded for a few frames after a cut o soglia – greater value for higher precision o moffset – greater value for higher precision 	<p>See defined profiles</p>
<p>-l on off maxdur <750></p>	<p>long shot configuration</p> <ul style="list-style-type: none"> o maxdur – a shot boundary is forced if shots gets longer than the given number of frames 	<p>Keep maximum value and occurrence will be negligible</p>

Table 11 - shotfinder command line options

4.3.2.5.2 *getstripeimages command line options*

<p>-f gray grayp 422 422p 420p</p>	<p>Tells which is the format of video input. It must match with decoder output.</p>	<p>- f 420p</p>
<p>-s <size></p>	<p>Tells geometry of video input, on Field Base. It must match with decoder output</p>	<p>-s 720x288 for standard resolution</p>
<p>-m (run mode) play afap</p>	<p>Run mode can be play (if input is real time from hardware) or as fast as possible (afap, when input is from a video file decoding)</p>	<p>-m afap</p>
<p>-j quality <75> smooth <0></p>	<p>jpeg quality configuration</p>	<p>Keep default</p>
<p>-R <step></p>	<p>If 0 tells to extract a column for each n frames Classical stripe image, time is horizontal</p>	<p>-R 2</p>
<p>-D <step></p>	<p>If 0 tells to extract a line for each n frames Time is vertical</p>	<p>- D 0</p>
<p>-S <spatial scale factor></p>	<p>Scale factor for spatial dimension.</p>	<p>Either -S 1 or -S 2</p>

Table 12 - getstripeimages command line options

4.3.2.6 Performances

The tools of the Gamp were originally developed for running real-time from live input with system that had quite poorer resources than those available today.

This implies that the Gamp can usually run faster than real time (one hour programme might be worked in 40 minutes). The CPU gets normally busy mainly because of the decoding process (ffmpeg).

Overall performances may also depend on :

- time required for accessing the material video on the EMS, which depends on configured protocols
 - it may require a copy of the file (file transfer) and then reading the file from the local file system
 - it may directly read the file made available by EMS (same system or mounted file system)
- time required for delivering key-frames and stripe-images to EMS

Parallel processing

On a very powerful machine it is conceivable to run more than one instance of the Gamp. However this has not been tested yet, and thus it is not recommended.

The easiest way to achieve massive-processing performance is to install the Gamp on several machines.

All the Gamp instances will inquire the same WorkFlow queue and work in the same time on different Working Units (Editorial Objects) without interferences.

4.3.3 Package 3

4.3.3.1 Operating system requirements

Speech to Text and Audio Structuring and Segmentation components run on GNU/Linux platform. Linux distributions successfully adopted comprehend RedHat9.0, Red Hat Enterprise 4, Fedora Core 3,5,6. With dual core hardwares 64 bit version is mandatory.

4.3.3.2 Hardware requirements

Speech to Text is capable to work with clusters of servers, even if a single host can perform the task. Each host needs at least 2 GB of RAM while the harddisk space required is around 3GB (most of the space required for the linguistic model). Gigabit lan connection is recommended .

The time required to perform the task scales almost linearly with the available cpu power with an upper bound related to the segmentation of the audio input and the management of the clusters.

As a rough indication a preprocessing time equal to play time can be achieved with four hosts dual processors 3.0 Ghz or two dual core dual processors.

4.3.3.3 Environment software requirements

A minimal linux installation is required. In particular is necessary

- perl 5.8.0 or later

- ssh 3.5 or later (in case of cluster architecture)
- ftp client (to put results) ??
- java runtime (version ?)

4.3.3.4 User guide

4.3.3.4.1 Installation

Consists in installing the core Automatic Speech Recognition software and the GAMP related part.

4.3.3.4.1.1 Automatic Speech Recognition

Just copy the entire tree called **protorai3** from the DVD (or network source) under local folder */opt/ITC-IRST*. In case another location is preferred is necessary to create a symbolic link.

E.g. : product copied in */mnt/store0/protorai3*, a link under *opt* and named *ITC-IRST* is necessary and can be created with a command like *"ln -s /mnt/store0/protorai3 /opt/ITC-IRST"*.

4.3.3.4.1.2 GAMP

All the GAMP stuff can be installed extracting the tarball archive *speech.gamp.tar.gz* in a predefined location (command *tar -xvzf speech.gamp.tar.gz*).

A subtree named *SpeechGAMP* is created containing all the stuff is necessary

4.3.3.4.2 Configuration

Other operations necessary to use the system includes for the ASR:

- setting of the correct language at system level (edit the file *"/etc/sysconfig/i18n"*)
- setting of the shared memory parameter to 1 GB (the usual way is to add in *"/etc/rc.local"* the command: *"/bin/echo 1073741824 > /proc/sys/kernel/shmmax"*)
- authorising ssh access from master towards slaves if a cluster architecture is deployed (for this purpose a specific utility is available)
- setting of the *GAMPHOME* environment variable according to the place where GAMP software has been set up (can be done adding the code line *"export GAMPHOME=<yourplace>/GampSpeech"* in the file *"/etc/rc.local"* executed at host bootstrap)
- setting of the cluster, specifying one from each line the ip addresses or hostnames of slave hosts in the file *\$GAMPHOME/etc/cluster.conf*

E.g. pool of 4 hosts *master.domain.it, Slave1.domain.it, Slave2.domain.it, Slave3.domain.it*

File *cluster.conf* residing on host *master.domain.it* at location *\$GAMPHOME/etc* will contain the lines:

```
Slave1.domain.it
Slave2.domain.it
Slave3.domain.it
```

- setting of the information necessary to contact the workflow manager and the essence manager editing appropriately the file *\$GAMPHOME/etc/madcore_connection.xml*

E.g

```
<MADCORE_CONNECTION>
```

```
<WFURL name="http://82.112.192.105:80/prestospace/services/WF"/>
<PSOURL name="http://82.112.192.105:80/prestospace/services/PSO"/>
<EMSURL name="http://82.112.192.105:80/prestospace/services/EMS"/>
<USERNAME name="presto"/>
<PASSWORD name="space"/>
</MADCORE_CONNECTION>
```

4.3.3.4.3 Usage

To run the GAMP application, just launch the `speech.gamp` (located in `$GAMPHOME/bin` directory) from the location `$GAMPHOME/var/running` of the master node.

As a mandatory parameter specify the target Core Platform queue name.

E.g. `cd $GAMPHOME/var/running`

```
./speech.gamp CA_speech.
```

The GAMP performs an infinite cycle polling the workflow on the specific queue each minute, it takes care of communication towards the platform (by mean of generic `gamp`) and locally invokes software components for Audio Structuring and Segmantation and Speech to Text. Finally translates results as MPEG7 snippets, fits them into the Edob structure and gives it back.

The software keep trace of actions via logs written into `$GAMPHOME/log` folder.

4.3.4 Package 4

4.3.4.1 Operating system requirements

The RAI ARCHIE-CAS + MMStructure GAMP suite run entirely on GNU/Linux platform. Developments, tests and implementations used in PrestoSpace have been deployed and run on RedHat 8.0/RedHat 9.0 distributions. In-house test platforms have been tested on Fedora Core 3 and Fedora Core 4 as well. RAI does not ensure reliability of installations made on other Linux distributions.

4.3.4.2 Environment software requirements

The ARCHIE-CAS package uses a number of third party software packages, that will be included as parts of the delivered software package. Here is a list of the main third party libraries and packages employed by the suite:

- Zope + Openflow (www.zope.org)
- Ncftp
- Java SDK (recommended release >1.4.2)
- Tomcat (www.apache.org)
- PostgreSQL
- Ffmpeg
- Motion JPEG Tools
- Netpbm

The MMStructure GAMP package depends on some basic data elaboration libraries, notably xsltproc and the GenericGAMP library delivered together with the Core Platform.

4.3.4.3 User guide

4.3.4.3.1 Installation

4.3.4.3.1.1 Introduction

ARCHIE-CAS comes with an installation script set that performs all the needed installation operations. The ARCHIE-CAS architecture is made of two principal sub-systems: a core subsystem and a process subsystem.

The core subsystem includes the Workflow management component (implemented with Zope and Openflow), the process orchestration management component (implemented with PostgreSQL and Tomcat), the monitoring applications (running on Tomcat), the data result repository, and the task management application for the Video Clip detection.

The process subsystem includes the feature extractors processes and the two statistical processes performing the video clustering and the video clip detection respectively.

You can install as many process subsystems as you want (one for each available hardware node in your dedicated hardware farm), but you should install only one core subsystem. The more process subsystems are installed in a deployed instance of ARCHIE-CAS, the higher level of scalability is obtained in the feature extraction task and in the statistical tasks.

Together with the main ARCHIE-CAS installation, you must install the corresponding GAMP, for integration with the MAD Core Platform. The installation of the GAMP is made through the deployment of a separate software package, which includes the binary distribution of the Core Platform Generic GAMP for Linux platforms.

4.3.4.3.1.2 Detailed Instructions

The RAI ARCHIE-CAS installer is a Perl application including the installation of all the needed CAS components. The user has to interact with the installer to obtain a correct CAS installation. All the question asked by the installer are made using the **package.xml** as guide for the installation.

The xml descriptor used by the installer includes the definition of each component part of the CAS application and its steps to complete setup, including software requirements needed by the module or node.

The installer follows two different ways to install elements, depending on whether they are modules or nodes. Modules are all elements that are not referable to measure nodes. Nodes are elements involving measurement software. For the module part you can choose which module to install for the node element.

The installer is only an interface launching individual installation scripts that set up each of the modules/nodes of the ARCHIE-CAS architecture. All the arguments specified in the package.xml for each setup step are passed to the relative script.

The Software requirements structure will be processed recursively installing the deepest level at first, then proceeding backwards until the first requirements is found. By this way it's possible to define dependencies between software.

4.4 Launch an installation

4.4.1.1.1.1 The ARCHIE-CAS installation batch

You have to follow a few simple steps to launch the installation. You do not need to configure the installer itself. The installer application is the file **cas_setup.pl**. After extracting the zipped tar archive you can launch, as root, the command

```
#perl cas_setup.pl
```

```
-----  
R&I Content Analysis System installation  
-----
```

```
Choose which component would you install:
```

- 1: Core FC4
- 2: Processes FC4_i386
- 3: Processes FC4_x86_64

```
Digit the number or 'a' for all component installation - 0 to exit:
```

At this point you have to choose which components to install, digit the number and continue.

4.4.1.1.1.2 ARCHIE-CAS core subsystem installation

By choosing the option no. 1 you get the next question and the installer shows the lists of module or nodes of the components selected.

```
-----  
Installing component: Core  
-----
```

```
Which module has to be installed?
```

- 1: workflow
- 2: database
- 3: queue
- 4: logs
- 5: notifier
- 6: storage

```
Digit the number or 'a' for all - 0 to exit:
```

Now you choose again the option no. 1 obtaining this output

```
-----  
Installing module: workflow  
-----  
-----
```

```

Verifying software requirements
-----
Start installation of the requirement: openflow editor
Start installation of the requirement: openflow
Start installation of the requirement: zope
Installing zope
    
```

--> STEP requirement: preinstall <--

Installation dir [/opt/cas]:

The installer starts the workflow component installation with a preliminary analysis of software requirements and shows you the dependencies. Then it starts with the deepest element required: "zope", and it asks you to set the first argument for the preinstall step. If you don't write anything the default value will be used (in this example: /opt/cas). Variables defined once are valid for the rest of the process.

From this point to the end of installation you will be prompted to insert only argument values, all the other actions are automatic.

4.4.1.1.1.3 ARCHIE-CAS processes subsystem installation

This section describes the installation of "Processes" components . This can be done selecting options 2 or 3 of the first selection list.

```

-----
Installing component: Processes
-----
Which module has to be installed?
1 Node Initialization
Digit the number or 'a' for all - 0 to exit: 1
    
```

Installing module: Node Initialization

```

-----
Verifying software requirements
-----
Start installation of the requirement: genUtils
Start installation of the requirement: nodeInit
Installing nodeInit
    
```

--> STEP requirement: installer <--

Processes user [vtools]:

After the installation of the module part of "Processes" component, which could include some software requirements, installer asks you to choose which node has to be installed

Which type of node has to be installed?

- 1: RegisterNode
 - 2: Delivery
 - 3: Statistic
 - 4: Measure
- Digit the number - 0 to exit:

By choosing a node type you could be prompted for input data about argument for installation of other software required by the node. You can install all the "family" by selecting the parent.

List of available processes for Measure node

- 1: AllColor
 - 2: HSV
 - 3: Hue
 - 4: Saturation
 - 5: Value
 - 6: Luminance
- 7: AllTemporal
 - 8: DispFrameDiff
- 9: AllTexture
 - 10: Contrast
 - 11: Directionality

Which process and its subprocesses has to be installed?
 (By choosing a parent element also children will be installed)

Digit the number - 0 to exit:

During installation of node you could be prompted for interaction exactly as described in the previous steps.

The installer doesn't produce log itself but the scripts do and can be found in **/tmp**, the filenames begin with prefix **cas**.

4.4.1.1.1.4 GAMP installation

To install the GAMP make the following steps:

- Create a default user for the GAMP operations
- Go to the chosen installation directory
- Unzip the GAMP software package

- Change directory to src
- Run make command

4.4.1.1.2 Recommended deployment architecture

Basically any ARCHIE-CAS software component could be distributed on an independent hardware node, exploiting the fact that components interact each other via TCP/IP. We anyway recommend that the following requirements should be met for a typical installation:

- Install the GAMP software, the core subsystem and data result repository on the same hardware node (Main node).
- Install the process subsystem on a set of independent dedicated hardware nodes (Data processing nodes).

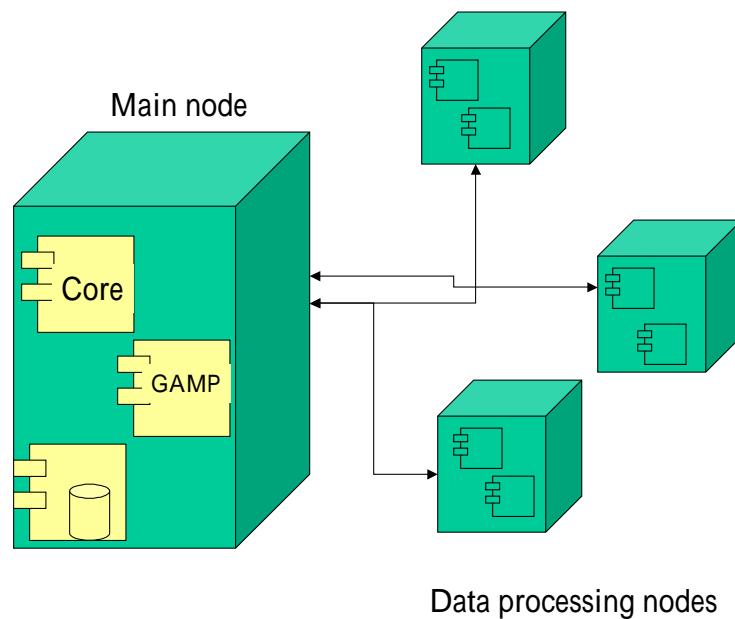


Figure 8. ARCHIE-CAS recommended deployment architecture

4.4.1.1.3 Test

To test whether an ARCHIE-CAS instance has been installed properly, run tests using the web page available at the following links:

http://CORESYSTEM:ZopePORT/CAS/prototipo-CAS/WebUtils/createExtProcessInstance_Form_zpt/

where CORESYSTEM is the name of the server on which the core subsystem is installed and ZopePORT is the TCP/IP port at which Zope has been configured to respond to HTTP requests.

Field	Meaning	Mandatory?
UID	The unique id of your instance	Yes
Customer	The name of the customer of the ARCHIE-CAS system	Yes. Always

Name	asking for a job.	end the customer name with an underscore character "_ "
Instance Title	The title of the Zope instance	No
Comments	Comments	No
Priority	Priority of the requested task	No (default=0).
FTP response server	The server where the results of the elaboration will be stored after completion (this must coincide with the data results repository).	Yes. Default is the server on which the core subsystem is installed.
FTP response user	The account name to access the data results repository. It is set during installation.	Yes. Default set during installation is "misuratore".
FTP response password	The password to access the data results repository. It is set during installation.	Yes. Default set during installation is "misuratore"
FTP response path	The path at which the results shall be deposited. It must coincide with the directory installation of the data results repository. It is set during installation.	Yes. Default is set during installation.
Delete param	Tells the orchestrator whether or not to erase some intermediate analysis results after completion.	No. Recommended to erase all results to save space.
Clip detection param	If checked, specifies that the statistical task to be run on the material is a video clip detection task. If not checked, a video cluster analysis is performed.	Yes.
Ref file param	This is a text area which must be formatted to indicate which video clips must be searched for detection on the defined material. The data input must be formatted following a comma separated format having the following semantics: DATA:= (CLIP_LABEL, DETECTION_THRESHOLD, FEATURE_LIST)+ CLIP_LABEL=string DETECTION_THRESHOLD=integer (between 0 and 100) FEATURE_LIST:=FEATURE_LABEL+ FEATURE_LABEL:=('hue' 'sat' 'contr' 'value' 'tem' 'lum' 'dir') ⁴ ' Feature labels cannot be modified, while clip labels must be chosen among the ones associated to the signatures during the reference clip selection phase (see 4.4.1.2.4).	Yes.
Source material type (pop up)	"Direct" selection means that you have to indicate the FTP access coordinates of the file you want to elaborate (host, path). This can be done also via uploading (or in-line editing) of an XML file containing a	Yes.

⁴ Hue, saturation, contrast, value, temporal activity, luminance and directionality respectively.

	<p>list of the files to be elaborated. If the list contains more than one file, the system will assume that the material is consisting of the concatenation of the files. An example XML file is provided in 4.4.1.1.3.1. By default all FTP media transfer operations are made by feature extraction processors using account name, password=ants,ants01. If you want to change them, please edit the configuration file named localConfig in the /INSTPATH/cas/core/vtools/batch directory of your installation.</p> <p>The "Acquisition" option is not running in the deployed version of ARCHIE-CAS. If you want to integrate this function, you have to develop a web service capable of responding with an XML descriptor like the one described in 4.4.1.1.3.1, using the following paradigm:</p> <pre> http://SERVICE-URL? canale=channel& startdate=YYYY-MM-DD& starttime=hh:mm:ss.cc& enddate=YYYY-MM-DD& endtime=hh:mm:ss.cc& limits=exact </pre> <p>The ARCHIE-CAS component in charge of orchestration (OpenFlow), will invoke this service and elaborate the XML formatted result, launching a distinct feature extraction task for each file, and performing the statistical analysis (video clustering or video clip detection) on the concatenated result features.</p>	
--	---	--

Once that an instance has been successfully created, a report page containing the name of the instance (INSTANCENAME) appears. From that moment on, you can monitor the completion of the instance by consulting the following page:

http://CORESYSTEM:ZopePORT/CAS/prototipo-CAS/Monitor/SummaryStatus_report

After that the job is completed, results can be browsed at:

http://CORESYSTEM/INSTPATH/cas/core/data/measures_db/INSTANCE_NAME/

where INSTPATH is the path at which the core subsystem has been installed.

4.4.1.1.3.1 XML input file example

```

<?xml version="1.0" encoding="UTF-8" ?>
<MediaInfo UID="myuid">
  <acq_channel>rete4</acq_channel>
  <MultimediaIntegrity>Contiguo</MultimediaIntegrity>
  <markin_gmt>2006-11-14 17:55:08.51</markin_gmt>
  <markout_gmt>2006-11-14 18:01:33.67</markout_gmt>
  <avilist>
    <avi file="/data/rete4/20061114/avi/rete4_20061114_17543043.avi">
      <info video_frames="3028" />
      <edl_info edl_from="952" edl_to="3028" />
    </avi>
    <avi file="/data/rete4/20061114/avi/rete4_20061114_17563155.avi">
      <info video_frames="3193" />
      <edl_info edl_from="0" edl_to="3193" />
    </avi>
    <avi file="/data/rete4/20061114/avi/rete4_20061114_17583927.avi">
      <info video_frames="3358" />
    </avi>
  </avilist>
</MediaInfo>
                
```

```

    <edl_info edl_from="0" edl_to="3358" />
  </avi>
  <avi file="/data/rete4/20061114/avi/rete4_20061114_18005355.avi">
    <info video_frames="3103" />
    <edl_info edl_from="0" edl_to="3103" />
  </avi>
  <MediaAccess>
    <host>ltocern304.servizi.rai.it</host>
  </MediaAccess></avilist>
</MediaInfo>

```

Note that files do not need to be AVI formatted necessarily, although the tag names are avi-orientated. Any available file format declared as decodeable by FFMPEG should be usable.

In order to complete the instance of work, data processing nodes must be up and running.

4.4.1.1.4 Running

To run the ARCHIE-CAS system you have to make sure that the core and process subsystem are up correctly, that the data result repository is accessible via FTP in read and write mode, and that the GAMP process is polling the queue on the Core Platform.

Normally, you do not need to play with the internals of the ARCHIE-CAS system during ordinary operations, unless you have to do deep debugging, because the GAMP client automatically creates instances, check results and integrate metadata in the EDOB document at the end of the operations. Most of the times, when things run smoothly, the monitoring pages are the only ones that need to be used by an administrator.

4.4.1.1.4.1 *Running feature extraction processes*

Feature extraction processes run on the dedicated Data processing nodes, and are launched automatically at the hardware node's startup. They are shell script performing workflow queues polling and launching data extraction processes. If you want to alter the number of running processes either modify the rc.local script on your installations, or perform a manual launch through the execution of the launch.sh script located in /INSTPATH/processes/vtools/ directory.

For PrestoSpace MAD, three extraction processes must run namely:

- colorTest_node_manager
- textureTest_node_manager and
- temporalTest_node_manager.

There are variations of the original ARCHIE-CAS scripts made on purpose for MAD. To use the launch script, please follow the example:

```
./launch.sh colorTest 1 NodeX localConfig textureTest 1 NodeY
localConfig temporalTest 1 NodeZ localConfig
```

Where node labels are those given by the user during installation.

4.4.1.1.4.2 *Running clustering and delivery processes*

Clustering and data delivery processes run on the dedicated Data processing nodes, and are launched automatically at the hardware node's startup. They are shell script performing workflow queues polling and launching data extraction processes. If you want to alter the number of running processes either modify the rc.local script on your installations, or perform a manual launch through the execution of the launch.sh script located in /INSTPATH/processes/vtools/ directory.

For PrestoSpace MAD two processes must run, namely:

- stat_node_manager

- `delivery_node_manager`

To use the launch script, please follow this example:

```
./launch.sh stat 1 NodeS localConfig delivery 1 NodeD localConfig
```

Where node labels are those given by the user during installation.

4.4.1.1.4.3 Running the GAMP

To run the GAMP application, you must launch the `segmenter.gamp` script under `$GAMPHOME/lib/Gamp/bin` directory of your main hardware node, specifying the corresponding Core Platform queue name. Example: `segmenter.gamp CA_mmStructure.`

Make sure that the file `$GAMPHOME/lib/Gamp/var/madcore_connection.xml` is set with the correct parameters for the access to the Core Platform services.

4.4.1.2 Configuration

4.4.1.2.1 Configuration of the Data processing nodes

No configuration is available for Data processing nodes.

4.4.1.2.2 Configuration of the video clustering application (ARCHIE-CAS)

The video clustering application delivered for the PrestoSpace MAD Unit has been tailored for the Documentation platform purposes, and therefore no particular configuration tool is available. However, here are some useful indications, suited for expert users who want to experiment different parameter settings than those wired in the delivered software packages. The statistics manager script launches the `stat_cluster_cluster` script with some fixed parameters, which can be changed in order to vary the clustering process behaviour.

In particular the following parameters can be varied:

- `<soglia_c_cluster>` minimum size of interval to be clustered, i.e. the threshold under which no further segmentation is done on video segments that are evaluated unstable during processing. If this threshold is raised, the overall process takes less time to finish, but aggregated video segments are less precise in boundaries.
- `<soglia_s_media>` maximum distance for aggregating an interval to a cluster, normalised to vary from 0 to 100, it represents the maximum average distance that two video segments must have in order to be clustered together. The higher is this threshold, the less precision in aggregation is obtained.

4.4.1.2.3 Configuration of the Multimedia Structure GAMP

The Media Structure GAMP is a separate software package from ARCHIE-CAS. It uses the ARCHIE-CAS results to extract multimedia structure information, which can either be used per se (MS-CAS.4) or as input data to the editorial part segmenter tool (EP-SHEF.1) or if it is itself used as an editorial parts segmenter tool (EP-CAS.1). It can be installed in any available hardware node connected both to the Core Platform node and to the ARCHIE-CAS core installation node. We recommend anyway to install the Multimedia Structure GAMP on the same node hosting the ARCHIE-CAS core component (see Figure 8).

GAMP jobs are downloaded from the Core Platform through the invocation of its web services. The multimedia structure job can have a flag (`flag_speaker_info`) indicating whether or not to take into consideration speaker labelling information provided by ASS-ITC.2 during elaboration. If the multimedia structure GAMP results are used as input data to the editorial segmenter, it is recommended to set this flag on. Otherwise, set this flag off to detect only the recurring visual structures or on to restrict the number of structures depending on the main speakers identified.

The GAMP doesn't come with any dedicated configuration tool. Here is a list of useful hints for expert users who might want to experiment different configuration parameters.

The script named `run_elaborazione.sh` located in `$GAMPHOME/bin`, launches the following scripts in sequence after having collected the results of the clustering elaboration and having the EDOB document available:

- `./clusterFilter.sh`. This script performs video clusters filtering, based on some clusters structural features which combination in a non linear filter determines whether or not a certain video shots cluster is potentially a story border discriminator. The following command line parameters can be varied:
 - o `selector_threshold`: it is the filter threshold. If this threshold is raised precision should be enhanced, otherwise (if lowered) recall should be enhanced. It is recommended not to vary this parameter out of a +- 10% of the current value (1e+15) if the multimedia structure detection software is providing input data to the editorial parts segmenter (§3.9.3.2). The threshold can be set to 0 if all clusters having a number of segments greater than `Max_excluded_score` are to be selected (i.e. the software is used as a pure multimedia structure detector).
 - o `Max_excluded_score`: is the number of video segments that a video cluster must at least have in order to be considered by the filter. Clusters having less or equal than `max_excluded_score` segments are discarded aprioristically.
- If the Core Platform job has the flag speaker "on", `./speakerFilter.sh`
 - o `Numfirst`: indicates how many speaker labels must be considered as main speakers for the video clusters pruning step. The first `numfirst` labels are chosen among the ordered set of speaker labels having decreasing temporal coverage of the EDOB timeline.
- If the Core Platform job has the flag speaker "on", `./clusterPrune.sh`
 - o `filter threshold`: it determines the covering percentage that a video shots cluster must have w.r.t. the main speakers selected at the previous step in order to be passed by the pruning filter.

4.4.1.2.4 Configuration of the video clip detection application (ARCHIE-CAS)

The video clip detection application can be extended via addition of new reference clip signatures. This can be done through a simple management interface accessible at:

<http://CORESYSTEM:TomcatPORT/app>

This interface allows you to define new references and new applications. A reference is a stream of low level features signatures associated with a segment of a video clip, whilst an application is a set of references plus the specification of which thresholds and which features to use for the clip detection task.

The data are stored under `/INSTPATH/cas/core/data/references_db/` on your core installation server under two subfolders named *signatures* and *applications* respectively. Signatures can be selected from a list of resources previously elaborated by the video clustering application through the use of the "New Reference" function. Applications can be built by collecting items from the available references through the use of the "New Application" function.

4.4.1.3 Monitoring

An enhanced monitoring system is provided bundled with any ARCHIE-CAS installation, accessible at <http://CORESYSTEM:TomcatPORT/CasLog>.

4.4.2 Package 5

4.4.2.1 Operating system requirements

The **EP-SHEF.1** is implemented as a standalone Java application that uses the SWT toolkit for the graphical interface. Consequently it runs on any operating system that can provide an implementation of Java 5 and SWT (this includes all versions of Windows, most Linux distributions and modern versions of MacOS X). The tool was developed and tested on the 32bit version of Ubuntu Linux using the JDK version 5 implementation provided by Sun.

4.4.2.2 Environment software requirements

The only software requirement is a working installation of a java virtual machine, version 5.

4.4.2.3 Installation

The software is distributed as a zip archive named *usfd-mad-software.zip*. This includes the implementations of 3 GAMPs, including the EditorialPartsSegmenter GAMP. The installation process consists of simply unpacking the archive into a convenient directory. The Editorial Parts Segmenter GAMP can be started by running the *editorial-parts-segmeneter.sh* (if running on Unix) or *editorial-parts-segmeneter.bat* (if running on Windows) start-up script. In most cases the script will be able to automatically identify the location of the Java installation. However, if the script cannot find a valid Java installation, or if a particular installation should be used (in the case of several versions co-existing on the same system), then the environment variable `JAVA_HOME` should be used to point to the top-level directory of the Java installation to be used.

Some of the files inside the archive need to have their executable bit set in order for the application to run. There is no standard way of encoding permissions in a zip archive, so the archive provided uses the algorithm defined by Info-Zip's implementation of the zip and unzip commands (which is the default implementation on many Unix platforms). If an incompatible tool is used to unpack the archive, the permissions will be lost and they need to be re-created: all the files in the `ant/bin` subdirectory as well as all the files with the extensions `.sh` in the root of the archive need to be made executable.

4.4.2.4 User guide

The Graphical Interface



Figure 4.9 Editorial Parts Segmenter GAMP Graphical Interface

When the tool is started, the graphical interface depicted in Figure 4.9 is shown. This allows the setting connection options (the URLs to the Documentation platform services as well as the user name and password to be used when connecting) and the starting and stopping of the GAMP. By default the interface contains the right values for connecting to the Documentation Platform managed by Eurix; these can be changed as required, before starting the GAMP. Once the values are set, the GAMP can be started by pressing the Start button. The tool will begin polling the WorkFlow service and will execute all the jobs that are queued for it. The GAMP can be stopped by pressing the Stop button – this will cause the GAMP to finish the current task (if any), disconnect from the Documentation Platform and stop polling the queue. The GAMP can then be shut down by closing the widow.

The terminal window that was used to start the tool will display log messages that can be used to monitor the status of the tool.

Sometimes it is useful, for debugging purposes, to process an EDOB from a local file rather than using the Essence and Metadata Storage service from the documentation platform. The tool allows this by using the options in the lower pane of the graphical interface: while the GAMP is stopped, use the Open button to load an EDOB file, then press Start. The GMAP will start running in local mode - the EDOB file will be processed and the resulting modified EDOB file will be saved in the same directory as the original one, with a modified name.

Configuration

The GAMP can be configured by means of an XML file located in the root directory that resulted from unpacking the distribution archive. The configuration file is called *editorial-parts-segmenter.xml* and it contains the default configuration values. These can be changed before starting the tool. The parameters that can be changed include the values for the *window* and *threshold* parameters, the XPath queries denoting the input segmentations to be used and the weights (confidence values) associated with the input segmentations. For exemplification, here are the default values used:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM
"http://java.sun.com/dtd/properties.dtd">
<properties>
<comment></comment>
<entry
key="segmentation1.xpath">//mad:EditorialObject//mp7:TemporalDecompo
sition[@criteria='multimedia
structure']/mp7:AudioVisualSegment</entry>
<entry key="segmentation1.weight">1</entry>
<entry
key="segmentation2.xpath">//mad:EditorialObject//mp7:TemporalDecompo
sition[@criteria='text level segments']/mp7:AudioSegment</entry>
<entry key="segmentation2.weight">1</entry>
<entry key="window">20000</entry>
<entry key="threshold">0.8</entry>
</properties>
```

The file uses the XML format defined by the Java language for representing property sets (sets of name/value pairs).

5 Terms and conditions of use

5.1.1 Package 1: Package 1: JRS media-analyse suite

The software contained in the package has been developed by JRS. The third party components that the package uses are either available as part of the operating system (such as the DirectX runtime) or do not require specific licensing; either the license of these components allows the reuse in products without limitations or JRS has acquired a development license for the component (which does not require runtime fees).

For use of the software package beyond the PrestoSpace project, JRS will license binary versions of the software. The license costs depend on the number of licenses in the installation at the customer's site. For bigger installations site licenses might be negotiated. Maintenance contracts on yearly basis are offered as an option.

5.1.2 Package 2: RAI Shotfinder

This package has been developed integrating third party open source software components with parts developed by RAI-CRIT. While the former parts are generally covered by GPL licenses, RAI-CRIT retains the full copyright of the latter part of the software package. RAI-CRIT makes it available for use to the PrestoSpace partners free of charge for the remaining duration of the project. Later uses will be object of separate agreements with interested parties.

5.1.3 Package 3: RAI Audio Analysis suite

The core tool has been developed by ITC-IRST (an italian public research center of the autonomous province of Trento) using their knowledge in the speech recognition area and in collaboration with RAI that provided the annotated material for bootstrap and training the system tailoring it for the newscast program domain.

RAI does not retain source code for the core part while has taken care of several aspect to improve reliability and system integration aspects.

Actually RAI and ITC-IRST are evaluating a commercial partnership about distribution, maintenance and evolution of the Audio Analysis Suite.

5.1.4 Package 4: RAI ARCHIE-CAS + MMStructure

This package has been developed integrating third party open source software components with parts developed by RAI-CRIT. While the former parts are generally covered by GPL licenses, RAI-CRIT retains the full copyright of the latter part of the software package. RAI-CRIT makes it available for use to the PrestoSpace partners free of charge for the remaining duration of the project. Later uses will be object of separate agreements with interested parties.

5.1.5 Package 5: University of Sheffield Editorial Segmenter

The software package has been developed by the University of Sheffield and makes use of only open source external tools or libraries (such as XML processing libraries). The University of Sheffield retains copyright of the source code but will make this tool available free of charge to all interested parts for a period of at least two years after the end of the project.

6 References

- [BPM] A.Messina, L. Boch, "[MAD Area Business Process Model](#)", MAD area working document⁵.
- [MADDF] A. Messina, L. Boch, "[MAD Area data format specification](#)", MAD area working document⁶.
- [MDS3] [State of the Art of Content Analysis Tools for Video, Audio and Speech](#) (D15.3), PrestoSpace Deliverable 15.3, 2005. URL:
- [DEL] [Delivery Models](#), PrestoSpace Deliverable 16.4, 2006. URL:
- [ARCH] The Documentation Platform for the MAD Factory, PrestoSpace Deliverable 18.1 [DEL3.1] [Models and Protocols for Preservation Process Management](#), PrestoSpace Deliverable 3.1, 2006.

5 This will be included in deliverable D15.2 "PrestoSpace documentation framework"

6 This will be included in deliverable D15.2 "PrestoSpace documentation framework"