 Deliverable  D15.2 MDS2

PrestoSpace Documentation Platform

**ABSTRACT**

This deliverable illustrates the Documentation Platform, which is a complex system, made of various automatic software tools, processors, and services integrated under a properly defined architecture, able to perform a complete documentation process of audio visual works, in the framework of the PrestoSpace Factory. The information extracted or produced during this process are collected and structured within an Editorial Object Document, in XML, which is delivered to a Publication platform implementing the fruition of the archive by means of search and retrieval activity.

**KEYWORDS**

Archive management, archive exploitation, editorial object, GAMP, documentation work flow, content analysis, semantic analysis, manual annotation, MAD XML Schema

**DOCUMENT HISTORY**

<table>
<thead>
<tr>
<th>Release</th>
<th>Date</th>
<th>Reason of change</th>
<th>Status</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td></td>
<td>First Draft – document structure</td>
<td>Living</td>
<td>Confidential</td>
</tr>
<tr>
<td>0.2</td>
<td></td>
<td>Chapters 5-10</td>
<td>Living</td>
<td>Confidential</td>
</tr>
<tr>
<td>0.3</td>
<td>22/10/2007</td>
<td>Annex A</td>
<td>Living</td>
<td>Confidential</td>
</tr>
<tr>
<td>0.4</td>
<td></td>
<td>Annotation Gamp</td>
<td>Living</td>
<td>Confidential</td>
</tr>
<tr>
<td>0.5</td>
<td></td>
<td>Usability tests</td>
<td>Living</td>
<td>Confidential</td>
</tr>
<tr>
<td>0.6</td>
<td>08/01/2008</td>
<td>Revision draft</td>
<td>Living</td>
<td>Confidential</td>
</tr>
<tr>
<td>1.01</td>
<td>25/02/2008</td>
<td>Disclosed as public document</td>
<td>Final</td>
<td>Public</td>
</tr>
</tbody>
</table>
# Contents Table

1. Document Scope ........................................................................................................... 3
2. Executive Summary ........................................................................................................ 3
3. Overview ......................................................................................................................... 3
4. References ....................................................................................................................... 4
5. Documentation business process model ........................................................................ 5
   5.1. Context analysis ........................................................................................................ 5
   5.2. Documentation .......................................................................................................... 14
6. Documentation model and format .................................................................................. 20
   6.1. Support to archive inventory .................................................................................... 20
   6.2. Support to documentation process .......................................................................... 23
   6.3. Metadata format reference ...................................................................................... 25
7. Platform architecture ..................................................................................................... 29
   7.1. Overview ................................................................................................................... 29
   7.2. Core Services .......................................................................................................... 29
   7.3. GAMPs ..................................................................................................................... 30
   7.4. Integration with PrestoSpace Orchestration ............................................................... 36
   7.5. Manual Annotation GAMP ....................................................................................... 36
8. Process design .................................................................................................................. 38
   8.1. Input ........................................................................................................................... 38
   8.2. Documentation workflow .......................................................................................... 38
   8.3. Outcomes .................................................................................................................. 40
9. Example of chains .......................................................................................................... 41
   9.1. Newscasts ............................................................................................................... 41
   9.2. News magazines ....................................................................................................... 42
   9.3. Documentaries ......................................................................................................... 42
   9.4. Others ....................................................................................................................... 43
10. Evaluation ...................................................................................................................... 44
    10.1. Annotation efficiency with/without automatic metadata extraction ....................... 44
    10.2. Retrieval performance of pure automatic metadata extraction ............................... 44
    10.3. Evaluation Criteria ................................................................................................ 45
    10.4. Usability tests on Manual Annotation GAMP ....................................................... 46
11. Conclusion ..................................................................................................................... 48
12. Glossary ......................................................................................................................... 49
13. Annex A ........................................................................................................................ 50
    13.1. MAD XML Schema .............................................................................................. 50
    13.2. Core platform definitions XML Schema .............................................................. 54
    13.3. Sample of Editorial Part XML fragment ............................................................... 55
1. Document Scope

This deliverable illustrates the Documentation Platform, which is a major component of the PrestoSpace Factory, a complete system for preservation, restoration, annotation and exploitation of the assets of audio visual archives.

2. Executive Summary

The documentation platform is a complex system, made of various automatic software tools, processors, and services integrated under a properly defined architecture, able to perform a complete documentation process of audio visual works. The information extracted or produced during this process are collected and structured within an Editorial Object Document, in XML, which is delivered to a Publication platform implementing the fruition of the archive by means of search and retrieval activity.

3. Overview

The main purpose of the Documentation Platform is to provide an advanced and mostly automated system for the documentation of audio visual archive works. The design of such a system implies the understanding of the archive management and exploitation processes, by means of case modelling, and the definition of a documentation model and an XML document format supporting the required documentation process.

The task of documentation is to produce various information elements useful or needed for the subsequent documentation steps and finally for the archive fruition. Several specialised tools may have to be employed and orchestrated depending on their availability and on the configuration of the documentation process itself. That’s why it has been deemed suitable to define a flexible and scalable platform architecture, made of a few core services and an assortment of processors, named GAMPs, easy to plug-in in a specified standard way and among which there are tools for content and semantic analysis, editorial segmentation, and manual annotation.

The process design includes the configuration of the documentation workflows, which are defined also according to the typology of audio visual work.

The issue of evaluation of the documentation process is also discussed. This is faced both in terms of efficiency and process cost comparison and in terms of retrieval performance. The human usability has been taken into account with respect to the manual annotation tool.
4. References


5. Documentation business process model

5.1. Context analysis

The present analysis is based on the assumption that the MAD area processes, and consequently the MAD system that will be implementing them, are closely related to the archive management processes, and eventually represent a commodity to better exploit the value already present in the archive. This means that the needs of the archive management domain must be taken into account as the starting points of the analysis.

Figure 1 - archive management processes

In Figure 1 the sub-processes that are relevant to the MAD activities are made manifest. The Archive Exploitation process is the objective to which the other sub-processes, as Digitisation and Restoration, are finalised. However, in order to actually use the archive content, an Archive Inventory process is necessary, as pointed out by the explicit dependency relation in the diagram. The inventory process is here intended as the way to obtain and maintain the information about which are the owned editorial assets and how they are physically realised by the audiovisual items stored in the physical archive.

To achieve this goal, some distinct and collaborating use cases are identified: the Editorial Object Inventory, the Editorial Collections Inventory and the Material Inventory, as showed in Figure 2.
Figure 2 - archive inventory process
This representation expresses a common situation: on one side, there is the need of finding which are the material instances realising a certain identified editorial object, whilst on the other side it is useful or necessary to know which are the editorial objects realised by certain identified material instances. Finally, the knowledge about the editorial collections (e.g. programme series) completes the scenario. The diagram shows this concept by means of a direct association between the material inventory process and editorial objects inventory process, expressed through the use of a collaboration. Furthermore, the editorial collections inventory is associated bi-directionally to the editorial object inventory process. This means that either the discovery of an editorial collection may give raise to a process of discovering which are the editorial objects that are part of the collection, or that by finding individual editorial objects there might be the possibility to infer the presence of editorial collections. Of course independent discoveries (of editorial objects or collections) remain possible.

An important issue needs to be pointed out here about the archive inventory activity. Usually, in real archives environments, programmes are likely split on more than one physical media. This is as more probable as the duration of the programme is longer. Another typical case, dual to the first one, contemplates that several programmes are recorded on a single media.

During the digitisation phase, in order to simplify the process, typical approaches tend to produce one file for each digitised physical media. However, to ensure a correct documentation process, the information about the mapping between the editorial object's timeline and the set of media timelines must be explicitly rendered and registered inside the documentation platform. This mean that the collaboration depicted in Figure 2 ("Association between editorial objects and materials") is charged of producing this mapping as well. More in detail, the mapping should include for each instant on the editorial object timeline the information about which media and at which time reference on the media is the actual realising source for that instant. This can be easily done by providing a simple compressed data structure based on the association of contiguous timeline segments respectively on the editorial object and on the media.

The Figure 3 shows an example of such a data structure in the case of multiple source media for a single editorial object. The exemplified data structure is general and it works for any other case of mapping too (e.g. multiple programmes on the same media), changing only the identification of either the media for a certain editorial object segment or the editorial object for a certain media segment.

An alternative scenario would be given if the digitisation process were configured to always produce a single file for each programme (or any equivalent editorial entity). In this case the timeline mappings would be the simplest possible, but this would also mean a “cut and paste” editing phase downstream to the digitisation. Besides, it could be required to keep the information of the mapping between the editorial objects and the legacy analog media, at least until these latter are preserved.

The Archive Exploitation use case is further detailed in Figure 4. Basically, the two main objectives of an archive user who is interested in exploiting the archive content are primarily those of retrieving the audiovisual items stored in the archive or retrieving the information thereby located in the same place.
While the former is quite a commonplace that doesn't need here particular clarifications, the latter represents a key point also in the perspective of defining some crucial aspects of any system delegated to implement a support for the archive exploitation functions. In fact, it gives account of the cases in which archive users are interested in obtaining pieces of knowledge without being directly interested in the audiovisual material somehow connected with it. For example knowing whether certain events happened or not, knowing about the content of a public speech of a famous person, and more in general making reports of certain aspects of social, arts, costume. From this point of view, the information that is living in the archive becomes itself an asset for the organisation owning the archive.

In a general way, both the retrieval functions can be distinguished in use and fruition. The use of an audiovisual item or of a piece of information consists in the actual employment of it in the context of production of something new (e.g. a new programme or a new multimedia service on the internet). This entails exportation from the archive of the audiovisual material or of the information in which the user is interested. On the other side, the fruition consists in the consumption of the audiovisual material or of the piece of information as a direct consequence of having retrieved it (e.g. preview of browsing quality material, reading of documentation and navigation in the documentation structure).

A last issue concerns the possibility that the digitisation process cannot produce the editorial object/media mapping due to the lack of sufficient knowledge about the editorial object content (e.g. old programmes, programmes with complex transmission structure and so on). It is supposed, conversely, that inside the MAD area processes the right human skills or automated processes could provide the proper means to recover this information back. Therefore, an explicit business process should be inserted as part of the documentation sub-process of the factory in order to take care of this aspect.
Figure 4 - archive exploitation process
Figure 5. Collaboration in archive exploitation
Some interesting collaborations arise from the actuation of the use cases and are shown in Figure 5:

- **Retrieving audiovisual assets by information.** This case can be viewed as the traditional audiovisual assets retrieval based on information acting as "metadata".
- **Retrieving information by audiovisual assets.** This case makes use of the audiovisual material as the carrier of information.
- **Retrieving audiovisual assets by audiovisual assets.** This case can be interpreted with the need to obtain audiovisual material that is similar to some other material regardless of the expressed content.
- **Retrieving information by information.** This case is a retrieval of information based on other information that act as "metadata".

**Definition of the MAD System mission**

With respect to the outlined context, the MAD system mission can be formulated as follows: **to provide concrete means through which to optimise the efficacy and efficiency of the exploitation process of an audiovisual archive.**

**5.1.1. Requirements and pre-conditions**

To fulfill its task, the MAD system must be supported by the Digitisation and Archive Inventory processes, as showed by Figure 6. This means that it is a pre-condition for the correct functioning of the MAD system that material sources are in digital form and that an archive inventory activity has been previously carried out. This includes that also the timeline mappings have been produced during the archive inventory process.

All throughout the MAD area process the update of the Archive Inventory shall be performed whenever either new Material are produced or new Editorial Objects are discovered and identified.
5.1.2. Basic MAD area use cases

The MAD system presents the basic use case structure depicted in Figure 7. A distinction is made between the core processes (Documentation and Publication) that are directly appointed to accomplish the system main mission, and the support processes, that act as utility function sets needed in order to ensure the flexibility (Process Configuration) and manageability (Content Management) of the system itself.

Moreover, among the support processes, we find also the Delivery which consists of the final activities of the MAD Area Process, after than the main objectives are achieved.

Figure 8 shows how the identified MAD system business processes are related to the realisation of the archive exploitation processes. The crucial MAD process is Publication. The Publication process, in turn, relies on the Documentation process which produces and provides the information base on which the archive retrieval functions can be actuated.
Figure 7 - overview of the MAD system business processes.

Figure 8 - use of MAD system processes to achieve archive exploitation
5.2. Documentation

The mission of the documentation process inside the MAD area system is to collect and produce information for efficient retrieve of audiovisual items and of information itself.

Figure 9. Overview of the MAD documentation process

In fact, as the diagram shown in Figure 9 illustrates, the use cases of which the documentation process is constituted reflect those defined for the retrieval process. In particular the following use cases are identified, that will be further described in the following sections:

- Complete identification information
- Identify and document editorial parts
- Get content information
- Enrichment with content related information

5.2.1. Complete identification information

- This use case deals with the collection and completion of the identification information for the editorial objects. This aspect can be further detailed as illustrated by
Figure 10, that includes the following sub-cases:

- Find titles
- Find contributions
- Find publications
- Find production information
- Find awards
- Find identifiers
It must be noticed that the information sources from which the above activities derive the required data can be internal to the MAD system, such as legacy information and the Materials themselves, as well as external publicly available sources (e.g. the web).

### 5.2.2. Identify and document editorial parts

This use case deals with the identification and subsequent documentation of the editorial parts of an editorial object. This implies that the documentation process can be applied recursively to the editorial parts that are identified for a certain editorial object. Different or specific criteria can be applied depending on the level of partition of the editorial object.
The above Figure 11 adds details to this use case. Particularly it clarifies the two different types of editorial parts that can be identified in this process:

- **Editorial constituent parts.** These objects represent the structures into which an editorial object can be decomposed from the point of view of the sequence of editorial items it is composed of. The editorial constituent parts of an editorial object must always be coherent segments of the editorial object’s timeline. A segment of an object is a coherent segment if its fruition has the same nature that the fruition of the whole object, i.e. all the tracks that are present in the decomposed object are also present in the segment considered in the partition and that all the tracks have the same decomposition. Figure 11 provides examples of some typical constituent parts (Sport items, documentary items and news items).

- **Editorial view elements.** These objects represent all the other possible structures into which an editorial object can be decomposed, provided that a well-defined editorial view is taken as a reference for the decomposition. It is not required for view elements to be coherent segments, e.g. an editorial view element may be defined with a segment having only the video track. Figure 11 provide examples of useful editorial view elements (shots and speeches).
5.2.3. Get content information

This use case deals with the collection and production of information deriving from the direct fruition of the audiovisual content and related to the meaning expressed by the editorial objects.

Figure 12 - get content information

As specified in Figure 12, the content information can be of different kinds, each of which originating a distinct use case:

- **Get text.** This use case is related to the production of the pieces of text that can be originated by the fruition of the audiovisual items (e.g. the text of speeches, the text displayed on video)
- **Get descriptions.** This use case deals with the production of pieces of information that specifically describe the audiovisual content and its meaning from different points of view (e.g. image and sound descriptions, programme synopsis, environmental descriptions).

- **Get classifications.** This use case deals with the production of pieces of information that classify the audiovisual content with respect to a set of categories.

- **Get topics.** This use case deals with the production of the pieces of information related to the entities that by various concerns constitute the subject of the editorial object or of one of its parts (e.g. persons, locations, works, objects, concepts).

- **Get content indexing.** This use case deals with the production of lists of links to meaningful access points through the timeline of the editorial object (e.g. key frame indexing, stripe image indexing, sound startup points).

Some of the use cases listed above, as Figure 12 points out, share the use of other use cases, in general dealing with the identification of particular entities, which can play different roles depending on the kind of content information that is being discovered.

### 5.2.4. Enrichment with content related information

This use case is related to the collection and production of the pieces of information that are related to the content expressed by the audiovisual items concerning a certain editorial object, but not directly derivable from the fruition of the audiovisual objects themselves and therefore obtained by external sources (e.g. the documentalist knowledge, the Internet, encyclopedias).
6. Documentation model and format

One outcome from the business process model analysis is the need to adopt a metadata format representation supporting not only the actuation of the documentation process but also the archive inventory process in a way suitable to the general orchestration of processes within a PrestoSpace Factory [D19.0.1 and/or D19.0.2].

6.1. Support to archive inventory

The Archive Inventory process is based on the knowledge of editorial objects, materials sources, and their associations. That relationship, named “Realisation”, is shown by the object diagram of Figure 13, the details of which are explained in Table 1.

![Figure 13 – object diagram of Material realisations](image)

The Editorial Object E1 was first realised by M1, which in fact has two source part links respectively with Ma and Mb, which have their respective sources in a couple of Storages (Sa and Sb) which could be actually two analog tapes. In the example it is figured that Ma and Mb are only partially the sources of M1, as the very beginning and the end are discarded. Moreover there is an overlap, that is a portion of the timeline of M1 which may come indifferently either from Ma or from Mb. As a matter of fact this is quite typical for a Programme too long to be recorded on a single tape, where the overlap was kept on purpose in order to permit a less critical playback of the whole Programme. Then the example supposes a digitisation process which leaded to the new material M2 for which the source is the single Storage S2, likely a computer file.

Table 1 - details on material realisation example

<table>
<thead>
<tr>
<th>Material</th>
<th>Source</th>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>Ma, Mb</td>
<td>Sa, Sb</td>
</tr>
<tr>
<td>M2</td>
<td></td>
<td>S2</td>
</tr>
</tbody>
</table>
6.1.1. Editorial Objects and Materials

Any editorial object instance must be identified all throughout the PrestoSpace context with a unique identifier. Its editorial timeline is specified by its duration, and it is basically classified according to which types of tracks are expected to be found in the related material instances.

Any material instance must be identified all throughout the PrestoSpace context by a UMID, which is an identifier created by the system which actually instantiated the material instance, according to SMPTE330M

The material timeline is specified by the duration, while the tracks information has to be fully specified.

The Realisation relationship is required to fulfill the inventory need.

In the Presto Space scenario it is expected that every Editorial Object instance have at least a realisation link to a Material instance.

The information about duration and track types have to be consistent.

It is not required for every Material instance to have a direct realisation link to an Editorial Object instance, provided that an indirect link can be established through other relationships, as explained ahead.

6.1.2. Editorial Object Collections

As reported at page 7, the archive inventory process is completed by the inventory of editorial collections, which may be programme series, episodes, anthologies, and so on. The data model and format support the collection providing room for complete identification, such as identifiers, titles, contributions, awards, and for referencing its components by means of the editorial object identifiers. Both directions of the relationship are supported so that for a given Editorial Object instance it is possible to know which is(are) the collection(s) it belongs to.

6.1.3. Materials sources

A Material instance can be originated either by another Material instance, through the source part relationship, or by a physical and concrete entity, here named Storage, which represents what is practically preserved within an archive, such as tapes and media files. For a given Material instance, the navigation through its source relationships will inevitably end on the Storage instances.

When a Material instance has a source link to a Storage instance, it means that by the proper use of that Storage (e.g. play the tape/file) it is possible to obtain exactly that Material. Any Storage instance must have one (and only one) source link to a Material instance.

Two Material instances may be linked one another through the source part relationship in which one instance acts as the source while the other one acts as the destination just as within a simple edit process.

When the source of a Material is a Storage it is implied that there are no gaps or overlaps in the material timeline.

When the sources of a Material are given by a set of links to other Material instances through the source part relationship, the occurrence of gaps and overlaps is possible. The overlap has to be considered as an interval of the material timeline where two (or more) sources provide exactly the same material, and it doesn’t matter which one will be actually used for a real play. The gap has to be considered as an interval of the material timeline where the source has not been identified.
6.1.4. Material derivation

Another important possible relationship between two Material instances is the Derivation, which permits to record information about the process of creation of a new Material instance based on a pre-existing instance.

There are several types of derivation and each of them can be associated to a specific set of information:

- **Transwrapping.** The new Material instance differs from the pre-existing one only because of modifications in how the sources (either material or storage) are organised, but the individual content elements are kept unchanged. It is assumed that the process does not produce a quality reduction.

- **Transcoding.** The new Material instance is obtained through decoding and re-encoding processes. In this case it may be assumed that the processes cause some kind of quality reduction. The processes may be accompanied by a modification in the organisation of the sources, as in the case of Transwrapping.

- **Restoration.** The new Material instance is obtained through processes which aim at the reduction of severe impairments which prevent the fruition of the pre-existing material instance. If the process is successful, the newly created Material instance can be considered a new master instance.

- **Digitisation.** The new Material instance is obtained by digitisation of an analogue material instance. This process is assumed to fix the quality to a level which can be maintained in the digital domain.

- **Analogue copy.** The new Material instance is created in the analogue domain. It is assumed that the resulting quality will be lower (not higher) than that of the pre-existing Material instance.
6.2. Support to documentation process

6.2.1. Complete identification

The original identification is made of a collection of various information elements:

- **Identifiers** – A set of string identifiers possibly of various types, which can also be given.
- **Titles** – A set of titles possibly of various types and languages, which can be further distinguishable for country.
- **Production information** – A set of properties of the Editorial Object including production date, production country, editorial control information, reference year.
- **Original Classification** – A set of classification properties which may be structured in a multi-dimensional manner in order to represent various aspects of the archive classification such as target audience, intention (education, entertainment, information, etc), format structure (bulletin, magazine, interview, commented event, etc), content or subject, etc.
- **Contribution** - a list of records which provide the role, the name of the played character (if relevant), and point to either the person or the organisation who/which contributed to the production of the editorial object.
- **Publication** - a list of records about publication events, referencing place, time and service of publication, optionally including publication score (audience rating).
- **Awards** - An Editorial Object may have “received”, directly or for a specific contribution, several awards, which are the grants of public approbation. Each of them is strictly related to the festival (the ceremony, the event) during which the award is assigned.

Figure 14 shows an example of object diagram on identification information together with the structuring suggested for the corresponding XML fragment of the Editorial Object document.
6.2.2. Editorial parts

Two Editorial Object instances can be linked by the relationship *Part*, where one of them plays the role of *container*, while the other one is the *contained*. It must be noticed that the latter has to be considered fully contained within the former and thus the starting point along the timeline of the container is sufficient to give the relative position correctly. The end of the contained Editorial Object is constrained to occur within the timeline of the container one.

An contained Editorial Object can be further classified in:

- *editorial constituent part* - when it has exactly the same track types as those of its container
- *editorial view element* – when its track types are a subset of those of the container editorial object.

Figure 15 shows an example of object diagram on editorial partitions together with the structuring suggested for the corresponding XML fragment of the Editorial Object document.

6.2.3. Content and content related information

To support the *get content* use case defined in §5.2.3 and showed in Figure 12 the Editorial Object has to be linked to a variety of *content information* elements. The general content information has to make reference to a specific interval on the editorial object timeline and to track types to which the information are pertinent.

In order to optimise the management of the links between content information and Editorial Objects, the content information elements have to be grouped under the *content information layers*, which record as *layer type* the criteria for its creation. The layer type has to belong to a defined enumeration.

The enrichment given by referencing content related sources, as defined by the use case defined in §5.2.4 is obtained by the relationship between the related document – such as an internet resource – and either the Editorial Object instance or the specific *content information* element.
Figure 16 shows an example of object diagram showing named entities (persons, organisation) which are related to the topic of editorial parts and links to resources sharing, partly, the same subject, together with the suggested structuring for the corresponding XML fragment of the Editorial Object document.

![Diagram of content related information for xml structuring](image)

**6.3. Metadata format reference**

The reference metadata format specifies a composite XML document, in which each individual component is expressed using the syntactic tools defined in the external standard (e.g. P_META EBU tech 3295, MPEG7) that is the most appropriate for the particular task.

The schematisation given by Figure 17 is summarised by the text of Table 2.

The complete XML Schema definition is given in Annex A.

<table>
<thead>
<tr>
<th>The global root element is an expressly defined structure directly connected to the Editorial Object business entity.</th>
</tr>
</thead>
<tbody>
<tr>
<td>The top level identification information, including production, publication and genre information, are expressed using P_META sets</td>
</tr>
<tr>
<td>The material realisation information is an expressly defined structure</td>
</tr>
<tr>
<td>The editorial partition and editorial views of the Editorial Object are expressed using an MPEG7 profile</td>
</tr>
<tr>
<td>All content related information is expressed using the same MPEG7 profile</td>
</tr>
<tr>
<td>Enrichment information is expressed using ad hoc structures</td>
</tr>
<tr>
<td>Ancillary data are realised using the same basic mechanism used for material realisation.</td>
</tr>
</tbody>
</table>

**Table 2 – defined criteria for metadata format structure**
Figure 17 - schematisation of the metadata format structure

Figure 18 - structure of the higher levels of the editorial parts/views and content information description
The top levels of the structure of the description which makes use of the MPEG7 profile are visualised in Figure 18, the leaf segments shown there contain content information description and may be further decomposed into sub-segments for certain types of description.

In Mpeg7 format each segment decomposition element is required to have a **criteria** attribute and for each segment the **StructuralUnit** element has a URI value which points to a term of structural unit classification schemes. Table 3 lists the assignments used with the documentation process.

<table>
<thead>
<tr>
<th>Semantics</th>
<th>Type of decomposition</th>
<th>Recommended criteria</th>
<th>Type of segment</th>
<th>Term Id</th>
</tr>
</thead>
<tbody>
<tr>
<td>root audiovisual segment into visual and audio part</td>
<td>MediaSource</td>
<td>modalities</td>
<td>AudioSegment VideoSegment</td>
<td>vis.programme</td>
</tr>
<tr>
<td>root visual segment into shots and transitions</td>
<td>Temporal</td>
<td>visual shots</td>
<td>VideoSegment</td>
<td>vis.shot vis.transition</td>
</tr>
<tr>
<td>root audio segment into speech segments</td>
<td>Temporal</td>
<td>speaker turns</td>
<td>AudioSegment</td>
<td>speaker.turn</td>
</tr>
<tr>
<td>shot into key frames</td>
<td>Temporal</td>
<td>key frames</td>
<td>VideoSegment</td>
<td>vis.keyframe</td>
</tr>
<tr>
<td>speech segment into parts</td>
<td>Temporal</td>
<td>syncs</td>
<td>AudioSegment</td>
<td>sync.words</td>
</tr>
<tr>
<td>root segment into editorial parts or an editorial part into further editorial parts</td>
<td>Temporal</td>
<td>editorial parts</td>
<td>AudioVisual Segment</td>
<td>editorial.part</td>
</tr>
<tr>
<td>editorial object timeline based on the audiovisual content structure (e.g. shot clusters, speaker clusters)</td>
<td>Temporal</td>
<td>multimedia structure</td>
<td>AudioVisual Segment</td>
<td>mm.structure .segment</td>
</tr>
<tr>
<td>root audio segment into audio spectrum image segments</td>
<td>Temporal</td>
<td>audio spectrum images</td>
<td>AudioSegment</td>
<td>aud. spectrumimage</td>
</tr>
<tr>
<td>root audio segment into audio volume image segments</td>
<td>Temporal</td>
<td>audio volume images</td>
<td>AudioSegment</td>
<td>aud. volumeimage</td>
</tr>
</tbody>
</table>

Table 3 – recommended values for “criteria” and “Structural Unit” reference
Figure 19 shows the organisation of the XML Schema defined to be used within the MAD area.

During the development of MAD Documentation platform the MAD XML Schema was importing and using P_META version 1.2, which was released at that time. Now the MAD XML Schema has been upgraded to import and use version 2.0, which carries significant improvements. The upgrade implies a corresponding upgrade of PSO, for the Editorial Object registration and the publication process, and a few aspects of the Documentation Core platform for the monitor interface.
7. Platform architecture

7.1. Overview

As explained in details in D18.1, the documentation platform architecture, the overview of which is given by Figure 20, contemplates a core system, providing work flow management and material and metadata storage services, surrounded by several processors (GAMPs) assigned to the accomplishment of specific documentation tasks.

The whole platform has to be considered under the control of the Presto Space Orchestrator (PSO) the interface to which are also provided by the Core component.

7.2. Core Services

7.2.1. Work flow management

The work flow management component holds a queue for each type of GAMP, and accepts in input the assignment of Editorial Object instances to defined job lists, containing the sequence of required processes, together with the desired options and parameters if any.

GAMPs have simply to inquire their queue in order to get a job in charge and then notify the result of the process in order to allow the progress along the work flow. This component also provides a monitoring interface which permits to track the status of both the queues and the Editorial Object instances.
7.2.2. Essence & metadata storage

Each generic processor (GAMP) is deemed to get an Editorial Object document in input and to produce an enriched version of it in output. The Essence & metadata storage component (EMS) provides a central service for maintaining the reference consistence copy of the Editorial Object documents during the documentation work flow. A CVS mechanism allows the GAMPs to check documents out and in, avoiding conflicts, although this approach prevents to run more than one processor on the same Editorial Object instance. Depending on the kind of performed process, the GAMP might require to work with some material instance in input – such as for video analysis – and / or to produce some kind of material in output – for instance key frames or stripe images. The EMS provides to the GAMPs the interfaces suitable to get access to the materials and for delivering newly produced materials. Once produced, the material instances get into the inventory process and become “read-only” for the subsequent activities.

7.3. GAMPs

The GAMPs are software units implementing tools for extracting information entities from various types of sources - such as audio, video, text, other documents – as represented in Figure 21. The final goal of the overall process is to fulfil the needs identified by the use case study of §5.2.3. In several cases the objective cannot be achieved through a single process because various intermediate results have to be produced separately and then collected for an inference process.

Figure 21 – generic documentation process building block
7.3.1. GAMP process structure

The basic behaviour of the generic GAMP towards the Core documentation platform is defined and it can be taken as reference for the implementation of any new GAMP. The installation requirement is that the work flow management component includes a queue for calling that GAMP from within a job list. The generic process structure is given in Table 4.

<table>
<thead>
<tr>
<th>The GAMP is deemed to run with a continuous loop:</th>
</tr>
</thead>
<tbody>
<tr>
<td>▪ on idle, inquire the queue for getting a new job</td>
</tr>
<tr>
<td>▪ check out the Editorial Object document</td>
</tr>
<tr>
<td>▪ get the material instances required for the process (if any)</td>
</tr>
<tr>
<td>▪ work</td>
</tr>
<tr>
<td>▪ insert the newly created material instances (if any)</td>
</tr>
<tr>
<td>▪ check in the modified Editorial Object document</td>
</tr>
<tr>
<td>▪ notify the completion of the job to the work flow manager</td>
</tr>
</tbody>
</table>

In case of failure during the process:

▪ undo the check out of the Editorial Object document
▪ notify the failure of the job to the work flow manager

Table 4 - work structure of the generic GAMP
7.3.2. List of GAMPs

Welcome

Purpose
Implemented by Rai, it aimed to provide video and audio material sources in the format suitable to the subsequent content analysis tools.

Input
Material instances at minimum broadcast quality.

Output
Material instances at broadcast quality.

Current status
This GAMP is replaced by the Transcoding services of the PrestoSpace Factory.

Speeches

Purpose
A content analysis processor, implemented by Rai, providing automatic speech recognition from audio material instances. It works with Italian and English languages (language must be indicated in input).

Input
Audio material in PCM_S16LE in WAV format.

Output
- Temporal segmentation with text of speech
- Temporal segmentation based on the speaker recognition

Current status
This GAMP is well tested. The performances on a generic language depend on a required preliminary training process and on the maintenance of an updated dictionary. It runs on Linux operating systems with at least 2GB of RAM. It is capable of subdividing the charge of work among a pool of systems reducing by that way the delay of accomplishment for the single job. When run on a single system, it took between 3 and 4 hours to work one hour long material.

Shots

Purpose
A content analysis processor, implemented by Rai, providing shot boundaries detection, key-frame & stripe images extraction from video material instances.

Input
Video material in broadcast quality

Output
- Temporal segmentation with shots
- Temporal segmentation with stripe images
- Key frames and stripe images pictures in JPEG format

Current status
This GAMP is well tested. It accepts in input various configurations permitting to adjust the performances in terms of optimising either the precision or the recall or keep a reasonable compromise. It runs on Linux operating systems and it takes typically about half an hour to work one hour long material.
Media analyse

Purpose
A content analysis processor, implemented by JRS, providing shot boundaries detection, key-frame & stripe images, camera motion detection, and low-level features extraction from video material instances.

Input
Video material in broadcast quality

Output
- Temporal segmentation with shots
- Temporal segmentation with stripe images
- Key frames and stripe images pictures in JPEG format
- Low level visual feature extraction from key frames
- Camera motion detection

Current status
This GAMP is well tested. It accepts in input various configurations permitting to adjust the performances in terms of optimising either the precision or the recall or keep a reasonable compromise. It runs on Windows operating systems and it takes typically about 5 hours to work one hour long material.

Multimedia structure

Purpose
A content analysis processor, implemented by Rai, providing temporal segmentation and clustering useful to a subsequent editorial segmentation from both video and audio analysis.

Input
Video and audio material in broadcast quality

Output
- Temporal segmentation for news programmes, based on either only shot structure or using both shot structure and speaker labelling information.

Current status
This GAMP is well tested on news programmes, optimisation on other programme typologies is in progress. It runs on Linux operating systems and it takes typically about one hour and a half to work one hour long material.
Lexical segmenter

**Purpose**
A content analysis processor, implemented by the University of Sheffield, providing temporal segmentation based on statistical analysis of text obtained in our case from automatic speech recognition tools and aiming to detect the change of content topic. Language independent.

**Input**
Temporal segmentation with text of speech

**Output**
- Temporal segmentation based on lexical analysis

**Current status**
This GAMP is partially tested on news programmes. Optimisation requires further training activity in progress. It runs on any modern Java-enabled operating system (including Linux, MacOS X and Windows) operating systems and it takes typically about five seconds to work one hour long material.

Editorial part segmenter 1st

**Purpose**
A tool, implemented by the University of Sheffield, providing a temporal segmentation of the Editorial Object instance into editorial parts through the analysis of other various temporal segmentation obtained by content analysis tools.

**Input**
Temporal segmentation

**Output**
- Editorial part segmentation

**Current status**
This GAMP performance strongly depends on the quality of the results of the preceding tasks. Roughly, best results are obtained by requiring higher precision if the preceding tools were set for higher recall and vice-versa. It runs on any modern Java-enabled operating system (including Linux, MacOS X and Windows) operating systems and it takes typically about one second to work one hour long material.

Editorial part segmenter 2nd

**Purpose**
A tool, implemented by Rai, providing a temporal segmentation of the Editorial Object instance into editorial parts using mainly the output of Multimedia structure, assuming that the Editorial Object is a news structured programme.

**Input**
Labelled Temporal segmentation

**Output**
- Editorial part segmentation

**Current status**
This GAMP is well tested on news programmes, optimisation on other programme typologies is in progress. It runs on Linux operating systems and it takes typically about twenty seconds to work one hour long material.
Semantic Analysis on English

**Purpose**
A semantic analysis processor, implemented by the University of Sheffield, providing named entity recognition, subject classification, retrieval of links to related content internet document from a document text in English language.
This tool has been used on the part of text obtained by a tool of automatic speech recognition corresponding to an identified editorial part.

**Input**
Text

**Output**
- Subject category
- References to Persons, Organisations, and Places
- Links to related sources

**Current status**
This GAMP performance strongly depends on the quality of the results of the preceding tasks. Subject classification depends also on preliminary training and on the adopted category set. It runs on any modern Java-enabled operating system (including Linux, MacOS X and Windows) operating systems and it takes typically about half an hour to work one hour long material.

Semantic Analysis on Italian

**Purpose**
A semantic analysis processor, implemented by University of Roma Tor Vergata, providing named entity recognition, subject classification, retrieval of links to related content internet document from a document text in Italian language.
This tool has been used on the part of text obtained by a tool of automatic speech recognition corresponding to an identified editorial part.

**Input**
Text

**Output**
- Subject category
- References to Persons, Organisations, and Places
- Links to related sources

**Current status**
This GAMP performance strongly depends on the quality of the results of the preceding tasks. Subject classification depends also on preliminary training and on the adopted category set. It runs on Windows operating systems and it takes typically about twenty minutes to work one hour long material.
7.4. Integration with PrestoSpace Orchestration

The PrestoSpace deliverables D19.0.1 and D19.0.2 explain that the Documentation platform constitutes a **Unit**, together with digitisation and restoration, of the **PrestoSpace Factory** and that the overall control component is the **PSO** (**PrestoSpace Orchestrator**).

Within the Documentation platform, the component in charge of interfacing with the PSO is that of Core Services. In particular the Work flow management considers the PSO as a client for inserting job lists regarding Editorial Object instances. Besides, other methods are provided to the PSO in order to provide progress monitoring functionality. Eventually it is important to notice that the selection of a process by means of defining a job list is a responsibility given to the PSO client component.

Moreover the PSO also provides EMS services for the whole PrestoSpace Factory, including the Documentation Units. However the GAMPs use the Essence & metadata storage services of the documentation core platform as a **proxy** for actually accessing the PSO EMS service.

At the completion of the documentation process the PSO has all the metadata and essence elements for continuing its task, for example delivering.

---

7.5. Manual Annotation GAMP

The Manual Annotation GAMP allows to view and validate results of automatic audiovisual and semantic content analysis. Furthermore it supports to manually structure content and add textual and semantic annotation. The tool uses MPEG-7 DAVP as input and output format. If content analysis results are available in the input MPEG-7 document, they can be displayed and modified. Otherwise annotation can start “from scratch”, i.e. by creating a new MPEG-7 document for the content to be annotated.

The architecture of the Manual Annotation GAMP uses a modular plugin system. All the components are plugins which use a framework for data access and to synchronise with other components. The components to be loaded are listed in the main application’s configuration file, so it is easy to extend the functionality by simply developing new plugins. In addition, within the GUI, components can be hidden/shown and moved freely, to adjust specifically to the users needs.

The Manual Annotation GAMP user interface contains a full featured video player combined with a number of timeline based views displaying structure information and content analysis results, a tree view of the content structure and controls for textual and semantic annotation.
The user interface of the Manual Annotation GAMP is displayed in Figure 22 which indicates the components numbered in braces and listed below:

- the video player (1)
- the main timeline (2) for navigation and specification of a temporal section of interest (local time period), and all views which display information of the local time period:
  - the shot editor timeline (6) for navigation and shot editing
  - the key frame view (7)
  - the stripe image view (8)
  - camera motion view (9)
  - speech to text (ASR) transcript view (5)
- the video structure tree view (3), which displays one or more high-level segmentations of the content and is also synchronized with the other views
- the semantic annotation view (4), which allows annotation of textual information and named entities on segment level.

![Figure 22 – Manual Annotation GAMP User Interface](image)

A plugin implements the integration with the PrestoSpace MAD Workflow and offers the functionality to get jobs for documentation, check-in documents and save the current documentation status in the EMS.
8. Process design

8.1. Input

The required input elements for the documentation process are an Editorial Object document – as the Editorial Object is the documentation working unit – and its material realisations in formats suitable for use by the content analysis tools. This implies that:

- the Editorial Object and its Materials are sufficiently well inventoried (see §6.1).
- Preservation/digitisation already performed successfully, as content analysis tools work on material in digital file format.

Within a PrestoSpace Factory all the inputs to a Documentation Unit are given by the PSO component, which is responsible for that.

The legacy archive metadata should be given by the Archive to the PSO Factory at the Editorial Object registration time and the Documentation Unit will assume that they are already included into the Editorial Object document. In particular it is expected to find some identification and publication information, which might be asked to be completed as described by the use cases of §5.2.1.

The details about the Editorial Object registration are explained in D19.0.1 and D19.0.2. Actually various practices may be recommended depending on the particular conditions of the Archive. We only mention here the following:

- The Archive may set up a mapping procedure to export the legacy information as Editorial Object document, which is then registered with the PSO
- The Archive may create the Editorial Object document by filling specific forms provided by the PSO interface
- A customised mapping procedure might be offered to the Archive by the PrestoSpace factory service provider organisation

8.2. Documentation workflow

The documentation workflow for a given Editorial Object instance is defined by the chain of GAMPs that make the process job list for that instance.

In D19.0.1 it is explained that the Archive defines the desired process as an Order which is submitted to the PSO and to which are assigned Batches of Editorial Objects. Concerning the documentation workflow, the PSO has to translate the services required within the Order into the most appropriate job list, solving the possible dependencies.

The criteria for the selection of specific tools should includes:

- Is the tool suitable for the kind of programme? Speech-to-text on Music programmes may achieve deceptive results.
- Consider also the quality of input material. If quality is too poor, automatic tools will drop their performance.
- Compare expectations and costs (in time/money/resources).
- Choose configurations suitable to your needs. In your case is it better to enhance the precision or the recall?
- Consider the best way of employing the human resources.
8.2.1. Content analysis tool selection

Content analysis is usually the first step.

Simple video indexing tools, providing key frames are considered interesting because of low cost and usefulness of output in the archive fruition process. However there are cases for which those tools are not appropriate or useful, for instance for interview programmes made of a single shot by a single camera. If more complex video analysis is required there is likely no need to run also the simple tool because the same output will be already included.

Complex video indexing tools are interesting when the output are desirable either for subsequent processes – e.g. temporal segmentation – or for search & retrieval process during the archive fruition.

Automatic speech recognition is very useful in the case of those kind of programmes where the tools reach a good enough performance because they provide words, places on the timeline, which can be the input of other processes – e.g. semantic analysis – and/or be simply used by search & retrieval process. Together with audio segmentation and clustering functionality, such as speaker segmentation makes the basis for further editorial segmentation.

Other tools provide more specific temporal segmentations, which do not give by themselves the identification of editorial parts, however they might be very helpful as input for that task.

8.2.2. Editorial segmenter tools

The editorial segmentation has to be considered for the following reasons:
- It contributes to providing a detailed description of the Editorial Object
- The Editorial Part itself can be considered in many cases – such as for the newscasts – the reference unit for the search & retrieval process.

The latter involves an important decision for the following archive fruition. Besides, this is a domain where human work and automatic tools can be used in various configurations:
- Automatic only
- Human correction on automatic inferences
- Human only, although helped by the output of other content analysis segmenters

8.2.3. Semantic analysis tools

The aim of semantic analysis is to collect information entities and knowledge to be used in the search & retrieval process. That’s why they have to work on the same reference units than for archive fruition.

Also for this kind of tools it is useful to consider how to integrate a human intervention, possible options are:
- Automatic only
- Automatic only but on editorial parts verified by human annotator
- Feedback from human annotators as improvement of training and setting
8.3. Outcomes

The delivery to the PSO from the Documentation Unit is automatically obtained by the completion of the job list.

In order to get newly documented instances into the circuit of the archive fruition, a delivery process has to be performed by the PSO towards the Archive.

Basically the following scenarios are possible:

- The Archive has got its own Archive fruition system and it will need simply to get back from the PrestoSpace Factory and import properly:
  - the enriched Editorial Object document
  - the newly created material instances (new digital master, browsing copies, key frames, etc)
  - optionally, the update of any knowledge base repository enriched by semantic analysis tools (see D19.0.1 for details)
- The Archive has got a PrestoSpace Publication platform (see D18.2). In this case the procedure of publication, which includes a process on the Editorial Object for its indexing with the search & retrieval system, is integrated.
- The Archive has got a PrestoSpace Turnkey System, in which case documentation and publication functionalities are integrated within the same system.
9. Example of chains

9.1. Newscasts

The case of newscast material was the most used within the tests performed along the MAD area work, because of the samples provided by both RAI and BBC and of the particular suitability for automatic metadata extraction. The most used and suggested chain is given in Table 5.

<table>
<thead>
<tr>
<th>GAMP or tool</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>shots or media analyse</td>
<td></td>
</tr>
<tr>
<td>speeches</td>
<td></td>
</tr>
<tr>
<td>multimedia structure and/or</td>
<td>depending on requirements of editorials</td>
</tr>
<tr>
<td>lexical segmenter</td>
<td>segmentation tool</td>
</tr>
<tr>
<td>editorial segmentation</td>
<td></td>
</tr>
<tr>
<td>semantic analysis</td>
<td>depending on language. All options</td>
</tr>
<tr>
<td>manual annotation</td>
<td>very optional</td>
</tr>
</tbody>
</table>

Table 5 - example of documentation chain for newscasts

The observation of results has pointed out that in many case a pure automatic annotation is possible in this case, provided that the newscast structure is regular enough and fit the editorial segmentation tool characteristics. For example a newscast with a single speaker in studio launching the news item headline followed by a news item non-studio contribution is the typical structure of many Rai samples.
9.2. News magazines

Some samples of news magazine programmes were also available during the tests of the documentation tools. Actually this class includes programmes quite different in terms of structure and this may imply a degree of arbitrariness when dealing with editorial parts.

Besides a single magazine can be either about a single subject, in which case the division into “parts” is smooth and poorly meaningful, or about well separate topics. The facts of having guests or active audience or being a live programme make difficult to determine a well structured decomposition, and may affect the performances of ASR tools and subsequent processes.

Therefore a suitable documentation chain could be as that of Table 6, where the manual annotation should be especially directed to the examination of the editorial parts obtained as draft by an automatic tool.

<table>
<thead>
<tr>
<th>GAMP or tool</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>shots or media analyse</td>
<td></td>
</tr>
<tr>
<td>speeches</td>
<td></td>
</tr>
<tr>
<td>multimedia structure and/or</td>
<td>depending on requirements of editorials</td>
</tr>
<tr>
<td>lexical segmenter</td>
<td>segmentation tool</td>
</tr>
<tr>
<td>editorial segmentation</td>
<td></td>
</tr>
<tr>
<td>manual annotation</td>
<td>correction of editorial segmentation</td>
</tr>
<tr>
<td>semantic analysis</td>
<td>depending on language. All options activated</td>
</tr>
</tbody>
</table>

Table 6 - example of documentation chain for news magazines

9.3. Documentaries

Also in the case of documentaries the pivotal question is: “is it a single subject homogeneous editorial object or is it a sequence of well defined parts for each of which there is a separate subject?”. The answer might be somewhat in the middle as well, such as when various topics are related among each other.
9.4. Others

It is difficult to set up a chain for working on programmes not sufficiently well known from the editorial and structure points of view. The main decision to be taken is that on the unit to be considered for the search and retrieve process. Also the suitability of the various automatic tools might be quite questionable.

A possible solution in this case is the “meta-chain” of Table 7, where a single shot detection and key-frame extraction tool is used in order to provide to human annotator a video index for easy access to material. The manual annotation is then directed to decide about which other automatic tools can be used and which kind of editorial segmentation is meaningful for each specific case.

<table>
<thead>
<tr>
<th>GAMP or tool</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>shots</td>
<td></td>
</tr>
<tr>
<td>manual annotation</td>
<td>or decision on subsequent part of chain</td>
</tr>
<tr>
<td>ny</td>
<td></td>
</tr>
</tbody>
</table>

Table 7 - documentation chain for unspecified programmes
10. Evaluation

10.1. Annotation efficiency with/without automatic metadata extraction

The general implied assumption is that a traditional annotation, without automatic metadata extraction, requires greater resources than an approach based on the process of automatic annotation tools for producing the same output and therefore the increase of efficiency can be easily evaluated.

Actually things are not as simple as that, because:

- the meaning of “traditional annotation” is ambiguous. For someone that means the use of a linear audio-video player for browsing the material and writing any single information on the keyboard. In fact basic improvements on such an annotation interface and non linear access to file recorded material produce a major increase of efficiency, while the annotation is still quite human based and traditional.

- automatic metadata extraction tools might be introduced progressively, from simpler and cheaper to sophisticated and more expensive. As an example, a “traditional annotation” simply supported by shot boundaries detection and key frames extraction can already take advantage in terms of efficiency.

- it is not granted that the product of the two annotation approaches has to be identical. The annotation process is performed with the aim of permitting later good search & retrieval process, however the strategy is not necessarily the same. Automatic analysis tools typically extract a great number of information entities, accepting an amount of errors in terms of precision and recall, relying on the fact that statistically the effect of those errors will be limited and compensated by the achieved retrievals. In fact the “cost per found information entity” tends to be negligible when the number of found information entities increases in the automatic extraction case. On the contrary, for the human annotation case the cost of finding an information entity is about constant, so the intuitive strategy is to optimise the quality of annotation with a much lower number of found information entities.

10.2. Retrieval performance of pure automatic metadata extraction

It is generally established the interest in evaluating the performances observed on the search & retrieve processes of the archive fruition in the case of a pure automatic metadata extraction system. This measurement doesn’t depend uniquely on the documentation tools, but it is also subject to the specific behaviour of the adopted search & retrieval techniques and tools and thus the system is evaluated as a whole.
10.3. Evaluation Criteria

An attempt to provide a formalisation of the evaluation criteria is given hereafter. In the diagram of Figure 23 a solely manual annotation system is compared with a hybrid annotation system, assuming the same documentation model. The quality of outcomes is detected at the output of the search & retrieve process, as a function of precision and recall, such as the \( F\text{-measure} \) (for the evaluation of recall a perfect knowledge about test material \( M \) is required).

The cost of production with the hybrid system is given by the sum of the costs of information extractions and the costs of human verifications. The formula given in Figure 24 shows how to identify in which cases the hybrid approach is less expensive than the manual one, provided that at least a minimum quality is achieved.

\[
\sum_{i=1}^{N} C_i^E + C_i^{CHK} + C_A \leq \sum_{i=1}^{N} C_i^H + C_M
\]

\( Q \geq Q_{min} \)

Unfortunately it isn’t easy at all to collect all the elements necessary to actually make the evaluation. The diagram of Figure 25, where for each quality on the horizontal axis is given a hypothetical corresponding cost for both manual (M) and automatic/hybrid (A) approaches, has been drawn assuming an intersection at \( (Q_b, C_b) \) for \( Q_b > Q_{min} \) and a greater cost of manual annotation for greater quality. Actually it might be quite the opposite (swap M and A).
A further difficulty is how to estimate the benefits of greater quality against the required cost increase.

10.4. Usability tests on Manual Annotation GAMP

The Manual Annotation GAMP was one of the tools in the general usability testing efforts that had to be tested always via a local installation, needing also results from the various automatic “metadata-generators” developed in the MAD work area. Therefore the tool could only be tested during the big testing workshop in November 2006 and during the 2nd testing phase in late spring/early summer 2007. To evaluate minor changes in the design after that period, two singular tests had been added in early fall 2007, but those haven’t been included in the overall analysis and results, as the test-sample was too small and the test-procedure took place under sub-optimal circumstances. Further information on the methodology (“Diagnostic Evaluation”) and the test-procedure can be found in D20.4.

Table 8 presents the average overall results of the questionnaire proposed to the users after the test, while the diagram of Figure 26 shows the improvement registered after the second test period.
<table>
<thead>
<tr>
<th>Q.ID</th>
<th>Question</th>
<th>Avg Res.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Has your overall expectation been fulfilled?</td>
<td>1.88</td>
</tr>
<tr>
<td>2</td>
<td>Has the tool supported you in fulfilling your task(s)?</td>
<td>1.63</td>
</tr>
<tr>
<td>3</td>
<td>Do you consider the tool concise and clear?</td>
<td>1.50</td>
</tr>
<tr>
<td>4</td>
<td>Would you use this (kind of) tool for your work, if available?</td>
<td>1.38</td>
</tr>
<tr>
<td>5</td>
<td>Is the display of the content analysis results useful?</td>
<td>1.56</td>
</tr>
<tr>
<td>6</td>
<td>Has the tool supported you in navigating through the video?</td>
<td>1.25</td>
</tr>
<tr>
<td>7</td>
<td>Please rate the structure visualisation</td>
<td>1.25</td>
</tr>
<tr>
<td>8</td>
<td>Please rate the structure editing</td>
<td>2.00</td>
</tr>
<tr>
<td>9</td>
<td>Please rate the named entities visualisation</td>
<td>2.13</td>
</tr>
<tr>
<td>10</td>
<td>Please rate the named entities editing</td>
<td>2.27</td>
</tr>
</tbody>
</table>

Table 8 – Average overall results of the post-test questionnaire (1=yes, 5=no)

The pleasant high ratings in Q1 and Q4 are proving a good acceptance of the tool, although the expectations (especially from the expert-users) were quite high (also caused by the successful user-workshops earlier in the project). A nearly sensational good rating was gained in Q6 and Q7, here the high effort in planning the structure of visualisation and navigation really paid off; the by way of comparison low rating in Q8 – Q10 were of course also influenced by the fact, that those parts are more familiar ones and have (workflow-wise and structure-wise) a much more prominent position in current comparable CMS’s and DB-tools.

![Figure 26 - Average Results Questionnaire Annotation GAMP](image)

Figure 26 - Average Results Questionnaire Annotation GAMP

The successful implementation of the inputs, recommendations and cognitions gained in the 1st test period is again demonstrated by the improvement of nearly all ratings, although during the 2nd test-period a sub-optimal hardware-setting had to be used (no dual-screen available for all test-procedures, caused by the use of laptops).
11. Conclusion

The Documentation Platform described in this document, conceived and realised within the PrestoSpace Project, proved to be capable of executing a tangible annotation work on audio visual documents so that they can be exploited through the publication system of the Archive organisation. Such work, based on the use of automatic software tools, can easily be set up to cover the case of massive digitisation for large collections, while being also appropriate for smaller batches. Moreover the resulting system is likely to provide future benefits: because the defined architecture allows flexible configuration, permitting to adjust the platform power to the customer expected throughput, and the adoption of any improved or future tool by using the defined GAMP framework, and because it defines and provides the whole document format set required for the platform integration. Further work would certainly carry additional results, in terms of optimisation of tools and of the documentation chain for a wider set of typologies, and regarding the issue of measuring the documentation efficacy and its impacts towards the retrieval performance.
## 12. Glossary

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVS</td>
<td>Concurrent versioning system</td>
</tr>
<tr>
<td>Editorial Object</td>
<td>Editorial Object. An audiovisual work, or an editorial constituent part of it, from the perspective of its artistic, communicable, and expressive aspects. It is identified as the PrestoSpace Factory Working Unit.</td>
</tr>
<tr>
<td>EMS</td>
<td>Essence and Metadata Storage</td>
</tr>
<tr>
<td>GAMP</td>
<td>Generic Activity MAD Processor</td>
</tr>
<tr>
<td>Job list</td>
<td>sequence of activities assigned to the documentation working units</td>
</tr>
<tr>
<td>MAD</td>
<td>Metadata Access &amp; Delivery</td>
</tr>
<tr>
<td>Material</td>
<td>An audiovisual entity which can be obtained by the playback of some audio/video recording. Not to be confused with Material Source, which in turn is a set of data which are the coded representation of audio-video recording. Material can be obtained by a Material Source by means of a decoding process.</td>
</tr>
<tr>
<td>Precision</td>
<td>The fraction of the documents retrieved that are relevant to the user's information need.</td>
</tr>
<tr>
<td>Recall</td>
<td>The fraction of the documents that are relevant to the query that are successfully retrieved.</td>
</tr>
<tr>
<td>UMID</td>
<td>The UMID is the Material Identifier as defined by SMPTE330M. In textual documents, as XML, it must be recorded as a string starting by &quot;0x&quot; and giving then the UMID value in hexadecimal (each byte is coded with a couple of characters in the range [0-9,A-F]). The UMID has 32 bytes with the a possible extension to total 64 bytes. The UMID has to be generated and provided by the entity emitting the Material instance.</td>
</tr>
</tbody>
</table>
13. Annex A

13.1. MAD XML Schema

Table 9 contains the full text of file MAD.xsd, which is the main MAD XML Schema document. In order to have the full XML Schema architecture, as described in Figure 19, the other required xsd components are:

- The CorePlatform definitions. See §13.2
- The MPEG7 DAVP Profile.
- The EBU PMeta2.0, the documentation of which is available from EBU-T3295v2. Current version of XML Schema and reference data documents are published at:
  - http://www.ebu.ch/metadata/cs/EBU_PMETAv020001_cs_p.zip

The directory structure and the schema location information used for importing the various XML Schema component have to match, check the import elements at the beginning of the XDS files.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.prestospace.org/MAD/DataModel/xml-schema"
  targetNamespace="http://www.prestospace.org/MAD/DataModel/xml-schema"
  xmlns:pmeta="urn:ebu:schema:pmeta:2007"
  xmlns:xmp="http://www.w3.org/2001/XMLSchema-instance">
  <xs:import schemaLocation="PMETA2.0/EBU_PMETAv020001p/EBU_PMETAv020001p.xsd"/>
  <xs:import schemaLocation="http://www.prestospace.org/MAD/CorePlatform/xml-schema"
    namespace="urn:mpeg:mpeg7:schema:2001"/>
  <xs:import schemaLocation="http://www.prestospace.org/MAD/CorePlatform/xml-schema"
    namespace="urn:ebu:schema:pmeta:2007"/>
  <xs:import namespace="http://www.prestospace.org/MAD/DataModel/xml-schema"
    schemaLocation="http://www.prestospace.org/MAD/DataModel/xml-schema"
    targetNamespace="http://www.prestospace.org/MAD/DataModel/xml-schema"
    elementFormDefault="qualified" version="2007-11-02"/>
  <xs:element name="EditorialObject" type="EditorialObjectType">
    <xs:annotation>
      <xs:documentation>The MAD area data format root element</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="EditorialObjectCollection" type="EditorialObjectCollectionType"/>
  <xs:simpleType name="msElapsedTime">
    <xs:union memberTypes="xs:nonNegativeInteger">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="determinate"/>
          <xs:enumeration value="indeterminate"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:union>
  </xs:simpleType>
  <xs:complexType name="EditorialObjectCollectionType">
    <xs:all>
      <xs:element name="Identifiers" type="pmeta:IdentifierDetailsType" minOccurs="0"/>
      <xs:element name="Titles" type="pmeta:TitleHistoryType" minOccurs="0"/>
      <xs:element name="LanguageInformation" type="pmeta:LanguageHistoryType" minOccurs="0"/>
      <xs:element name="Contributions" type="pmeta:ContributionType" minOccurs="0"/>
      <xs:element name="Awards" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Award" type="AwardType" minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Synopsis" type="xs:string" nillable="true" minOccurs="0"/>
    </xs:all>
  </xs:complexType>
</xs:schema>
```
<xs:element name="SourceStart"><xs:complexType><xs:simpleContent><xs:extension base="msElapsedTime"><xs:attribute name="indexType" type="xs:short" use="required"/></xs:extension></xs:complexType></xs:element>
<xs:element name="DestinationStart" type="msElapsedTime"/>
<xs:element ref="Material"/>
<xs:element ref="Derivation" minOccurs="0"/>
<xs:element name="UMID" type="UMIDType" use="required"/>
<xs:element name="MaterialRef"/>
<xs:attribute name="UMIDType" use="required"/>
<xs:attribute name="qualityLevel"/>
<xs:attribute name="Broadcast"/>
<xs:attribute name="Browsing"/>
<xs:element/>
<xs:complexType>
<xs:simpleType name="UMIDType"/>
<xs:complexType name="MaterialType"/>
<xs:complexType name="AnalogueCopyType"/>
<xs:complexType name="DigitisationType"/>
<xs:complexType name="RestorationType"/>
<xs:complexType name="TranscodingType"/>
<xs:complexType name="TranswrappingType"/>
<xs:complexType name="FileTyp" ebu:FileFormatCS (should be xs:anyURI)"
<xs:complexType name="StorageType"/>
<xs:complexType name="DerivationType"/>
<xs:complexType name="MaterialType"/>
The following types are defined for integration in the MPEG7 headers:

```
<xs:element ref="cp:Accesses" minOccurs="0"/>
</xs:sequence>
<xs:attribute name="name" type="StorageNameType" use="required"/>
<xs:attribute name="nameIssuer" type="StorageNameIssuerType"/>
<xs:attribute name="storageType" type="xs:anyURI"/>
</xs:element>
</xs:complexType>
```

```
<xs:complexType name="StorageNameType">
<xs:restriction base="xs:string"/>
</xs:complexType>
```

```
<xs:complexType name="StorageNameIssuerType">
<xs:extension base="mp7:HeaderType" minOccurs="0">
<xs:element name="Contributor" type="pmeta:ContributorDetailsType" minOccurs="0"/>
</xs:extension>
</xs:complexType>
```

```
<xs:complexType name="AdvisorType">
<xs:complexContent>
<xs:extension base="mp7:HeaderType" minOccurs="0">
<xs:element name="EventStart" type="pmeta:TimestampType" minOccurs="0"/>
<xs:element name="Place" type="pmeta:AddressType" minOccurs="0"/>
<xs:element name="Contributor" type="pmeta:ContributorDetailsType" minOccurs="0"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
```

```
<xs:complexType name="UMIDHeaderType">
<xs:complexContent>
<xs:extension base="mp7:HeaderType" minOccurs="0">
<xs:element name=" UMID" type="UMIDType" use="required"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
```

```
<xs:complexType name="EditorialObjectHeaderType">
<xs:complexContent>
<xs:extension base="mp7:HeaderType" minOccurs="1">
<xs:element ref="EditorialObject"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
```

```
<xs:complexType name="LabelHeaderType">
<xs:complexContent>
<xs:extension base="mp7:HeaderType" minOccurs="1">
<xs:element name="Label" type="xs:string"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
```

```
<xs:complexType name="AncillaryDataType">
<xs:complexContent>
<xs:extension base="mp7:HeaderType" minOccurs="0">
<xs:element ref="Material" maxOccurs="unbounded"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
```

```
<xs:complexType name="AncillaryData" type="AncillaryDataType"/>
```

```
<xs:complexType name="CharacterOffsetMediaTimeType">
<xs:complexContent>
<xs:extension base="mp7:MediaTimeType" minOccurs="0">
<xs:element name="Offset" type="xs:nonNegativeInteger"/>
<xs:element name="Count" type="xs:positiveInteger"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
```

![Table 9 - MAD XML Schema document](image-url)
13.2. Core platform definitions XML Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:dm="http://www.prestospace.org/MAD/DataModel/xml-schema"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://www.prestospace.org/MAD/CorePlatform/xml-schema"
    elementFormDefault="qualified" attributeFormDefault="unqualified" version="2007-01-15">
    <xs:import namespace="http://www.prestospace.org/MAD/DataModel/xml-schema"
        schemaLocation="MAD.xsd"/>
    <xs:attributeGroup name="fileAttributes">
        <xs:attribute name="md5_checksum" type="xs:string" use="optional"/>
        <xs:attribute name="num_bytes_qty" type="xs:positiveInteger" use="optional"/>
    </xs:attributeGroup>
    <xs:element name="WorkingUnit" xs:complexType>
        <xs:sequence>...
    </xs:complexType>
    <xs:element name="Materials" xs:complexType>
        <xs:sequence>...
    </xs:complexType>
    <xs:element name="MaterialExchange" xs:complexType>
        <xs:sequence>...
    </xs:complexType>
    <xs:element name="Accesses" xs:complexType>
        <xs:choice>...
    </xs:complexType>
    <xs:element name="Service" xs:complexType>
        <xs:all>
            <xs:element name="Protocol" xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:enumeration value="ftp"/>
                    <xs:enumeration value="http"/>
                    <xs:enumeration value="smb"/>
                    <xs:enumeration value="ssh"/>
                    <xs:restriction type="xs:dateTime" use="required"/>
                </xs:restriction>
            </xs:element>
            <xs:element name="Host" xs:complexType>
                <xs:sequence>...
            </xs:complexType>
            <xs:element name="Context" xs:complexType>...
        </xs:all>
    </xs:complexType>
    <xs:element name="Shipment" xs:complexType>
        <xs:all>...
    </xs:complexType>
    <xs:element name="Shipments" xs:complexType>
        <xs:all>...
    </xs:complexType>
    <xs:element name="Packages" xs:complexType>
        <xs:all>...
    </xs:complexType>
    <xs:element name="Manifest" xs:complexType>
        <xs:all>...
    </xs:complexType>
    <xs:element name="ShipmentId" xs:complexType>
        <xs:restriction base="xs:string"/>
    </xs:complexType>
    <xs:element name="id" type="xs:string" use="required"/>
    <xs:element name="deliveryDate" type="xs:date" use="optional"/>
</xs:schema>
```

Table 10 - Core Platform XML Schema Document
13.3. Sample of Editorial Part XML fragment

In the tables presented in this section it is possible to find a commented example of the XML fragment corresponding to the audiovisual segment of type editorial part, taken from an Editorial Object document. Table 11 shows the main node and its major components which will be expanded within the subsequent tables:

- **EditorialObject** node, within the mpeg7 header, enlarged in Table 12, contains the identification information (see Figure 14) relevant for the editorial part case
- The **CreationInformation** node, see Table 13, is used for recording **categories** and **related materials**
- The **Semantic** node, is used for named entities, details in Table 14

---

**Table 11 – XML fragment of an editorial part segment example**

The example of Table 12, has been made, assuming to have only the contributors to the specific news item – author(journalist), camera and editing operators – and a synopsis which is coming from the archive itself.

---

**Table 12 – example of EditorialObject XML fragment**
Table 13 contains the information which are expected from a semantic analysis tool as subject category, which is simply given by the value href of the Genre element, and related sources, structured by the RelatedMaterial element. The example shows the required syntax which make the fragment compliant to the mpeg7 specification.

We can have from zero to unbounded Related Materials. In case RelatedMaterial (e.g. web document) is given, the MediaLocation is necessary. The information given by Title, Abstract, Subject can be used by the EdobViewer in order to describe the link to the resource into its anchor. An expected behaviour is to pick the first available from:

- Title
- First n characters of Abstract string
- First n characters of Subject string
- the MediaURI (this is always available)

The information can be used by EdobViewer also for giving more details about related stuff and in any case for indexing the editorial part in the most suitable way.

```
<mp7:CreationInformation><mp7:Creation><mp7:Title/></mp7:Creation>
<mp7:Classification/>
<!--Following element record the category(ies) Coding (attribute href) needs to be better defined.-->
<mp7:Genre href="urn:x-prestospace-mad:cs:GenreCS:2005:CRO" type="main"/>
</mp7:Classification>
<mp7:RelatedMaterial>
</mp7:MediaLocator>
<mp7:CreationInformation><mp7:Creation><mp7:Title>Maltempo. Crolla un ponte in Puglia: quattro morti e due dispersi. Deraglia treno: 22 feriti</mp7:Title></mp7:Creation>
<mp7:Classification><mp7:Subject><mp7:FreeTextAnnotation>Maltempo Bari Crollo Ponte Morti Puglia Deraglamento Treno Eurostar Taranto-Milano Protezione Civile Bertolaso Prefettura</mp7:FreeTextAnnotation></mp7:Subject></mp7:Classification>
</mp7:CreationInformation>
<mp7:RelatedMaterial>
<mp7:MediaLocator><mp7:MediaUri>http://www.lagazzettadelmezzogiorno.it/GdM_cronache_NOTIZIA_01.asp?IDCategoria=273&amp;IDNotizia=146039</mp7:MediaUri>
</mp7:MediaLocator>
<mp7:CreationInformation><mp7:Creation><mp7:Title>Piogge torrenziali: quattro morti in Puglia</mp7:Title></mp7:Creation>
<mp7:Abstract><mp7:FreeTextAnnotation>Un ponte è crollato nella notte a Cassano Murge, centro in provincia di Bari, risucchiando le auto che vi transitavano con i loro passeggeri, mentre un giovane è stato trascinato dal fango lungo il canalone di Bari sino al mare. All'altezza di Acquaviva delle Fonti un treno è deragliato per le forti piogge. Il capo della Protezione civile, Guido Bertolaso è in prefettura a Bari per seguire di persona la gestione dei soccorsi</mp7:FreeTextAnnotation></mp7:Abstract>
</mp7:Creation>
<mp7:Classification><mp7:Subject><mp7:FreeTextAnnotation>CROLLA PONTE; TRAVOLTA FAMIGLIA IN AUTO. AUTO PORTA VIA DAL FANGO NEL CANALONE, UN MORTO A BARI. DERAGLIANO CARROZZE EUROSTAR IN PUGLIA, FERITI</mp7:FreeTextAnnotation></mp7:Subject></mp7:Classification>
</mp7:CreationInformation>
<mp7:RelatedMaterial>
</mp7:CreationInformation>
```

Table 13 – example of CreationInformation XML fragment
The example of Table 14 is made expressly for giving a syntax sample for each type of named entity which is expected to be found by the semantic analysis tool, and specifically:

- Place
- Organisation
- Person

The \textit{Semantic} node wraps all the named entity objects for each of which there is a \textit{SemanticBase} element.

The attribute \texttt{xsi:type} should indicate the type of named entity, at the \textit{SemanticBase} level for Place, and at the \textit{Agent} level for Person and Organisation (both \textit{AgentObjects}).

The optional \textit{Mask} element is defined to provide the position of the named entity into the text coming from ASR. Implicitly it provides the number of occurrences of the named entity. The value of the \textit{MediaTimePoint} must be the same of one \textit{MediaTimePoint} used for the \textit{TemporalSegmentation} of text [\texttt{@criteria='syncs']
<mp7:SemanticBase xsi:type="mp7:AgentObjectType">
<!-- This named entity is an Organisation. The attribute xsi:type is REQUIRED but not sufficient for identifying the type of NE -->
<mp7:MediaOccurrence>
<mp7:MediaLocator><mp7:MediaUri>urn-x:prestospace:mad:KIM</mp7:MediaUri></mp7:MediaLocator>
<mp7:Mask xsi:type="mp7:TemporalMaskType">
<mp7:SubInterval xsi:type="CharacterOffsetMediaTimeType">T00:04:30:00F25</mp7:MediaTimePoint><Offset>3</Offset><Count>20</Count></mp7:SubInterval>
</mp7:Mask>
</mp7:MediaOccurrence>
<!-- Element Agent and its attribute xsi:type are REQUIRED in order to know which type it is between Person and Organisation -->
<mp7:Agent xsi:type="mp7:OrganizationType">
<!-- Element Name is not required and may be ignored by the publication process (unless all other names are missing) -->
<mp7:Name>ferrovie dello stato</mp7:Name>
</mp7:Agent>
</mp7:SemanticBase>

<mp7:SemanticBase xsi:type="mp7:AgentObjectType">
<!-- This named entity is a Person. The attribute xsi:type is REQUIRED but not sufficient for identifying the type of NE -->
<mp7:MediaOccurrence>
<mp7:MediaLocator><mp7:MediaUri>urn-x:prestospace:mad:KIM</mp7:MediaUri></mp7:MediaLocator>
</mp7:MediaOccurrence>
<!-- Element Agent and its attribute xsi:type are REQUIRED in order to know which type it is between Person and Organisation -->
<mp7:Agent xsi:type="mp7:PersonType">
<!-- Here elements "Name/GivenName" are MANDATORY according to mpeg7 Schema. They might be empty and may be ignored by the publication process (unless all other names are missing) -->
<mp7:Name><mp7:GivenName>Gianni Morandi</mp7:GivenName></mp7:Name>
</mp7:Agent>
</mp7:SemanticBase>
</mp7:Semantic>

Table 14 – example of Semantic XML fragment