MeMAD Deliverable

D6.7 – Specification of the data interchange format, final version

Version 1.0

Grant Agreement number	780069
Action Acronym	MeMAD
Action Title	Methods for Managing Audiovisual Data: Combining Automatic Efficiency with Human Accuracy
Funding Scheme	H2020-ICT-2016-2017/H2020-ICT-2017-1
Version date of the Annex I against which the assessment will be made	8.5.2019
Start date of the project	1.1.2018
Due date of the deliverable	31.03.2020
Actual date of submission	22.04.2020
Lead beneficiary for the deliverable	Limecraft
Dissemination level of the deliverable	Public

Action coordinator's scientific representative
Prof. Mikko Kurimo
AALTO – KORKEAKOULUSÄÄTIÖ, Aalto University School of Electrical Engineering, Department of Signal Processing and Acoustics
mikko.kurimo@aalto.fi

MeMAD project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 780069. This document has been produced by theMeMAD project. The content in this document represents the views of the authors, and the European Commission has no liability in respect of the content

Authors in alphabetical order		
Name	Beneficiary	e-mail
Karel Braeckman	Limecraft	karel.braeckman@ limecraft.com
Simon Debacq	Limecraft	simon.debacq@ limecraft.com
Harri Kiiskinen	YLE	harri.kiiskinen@yle.fi
Nico Oorts	Limecraft	nico.oorts@ limecraft.com
Lauri Saarikoski	YLE	lauri.saarikoski@yle.fi
Raphaël Troncy	EURECOM	raphael.troncy@ eurecom.fr
Wim Van Lancker	Limecraft	wim.vanlancker@ limecraft.com
Dieter Van Rijsselbergen	Limecraft	dieter.vanrijsselbergen@ limecraft.com
Maarten Verwaest	Limecraft	maarten.verwaest@ limecraft.com
Kim Viljanen	YLE	kim.viljanen@yle.fi

Document reviewers		
Name	Beneficiary	e-mail
Tiina Lindh-Knuutila	Lingsoft Language Services	tiina.lindh-knuutila @lingsoft.fi
Jorma Laaksonen	AALTO University	jorma.laaksonen @aalto.fi

Document revisions			
Version	Date	Authors	Changes
0.1	15/03/2020	Dieter Van Rijsselbergen	Initial version with updates to user stories inclusion of implementation epics and component requirements.
0.2	06/04/2020	Dieter Van Rijsselbergen	Updates to various sections on user stories, implementation epics components, and prototype.
0.9	08/04/2020	Dieter Van Rijsselbergen	Processed internal review comments.
1.0	19/04/2020	Dieter Van Rijsselbergen	Final updates to exchange formats and document formatting.

Abstract

This deliverable defines the functional and non-functional requirements of the MeMAD prototype system, based on input concerning the tools developed in WP2, WP3, WP4 and WP5 and based on the project's overall use cases from which many requirements are defined.
This final version of the specification of the data interchange format updates the methodology that is followed to construct the project's prototype and refines the user stories and functional requirements of the MeMAD prototype system and its underlying components, based on the first two evaluation rounds of the prototype. It also defines the completed proposal for the various metadata exchange formats to be used between MeMAD components and the integrated platform. Finally, this document suggests updated evaluation criteria to determine the performance of the prototype system across the various use cases it implements.

Contents

1 Introduction

This deliverable represents the final set of results of work done on task T6.1, formally named the "*Specification of the data interchange formats*", but also includes preparatory work done to reach this outcome, defining overall prototype requirements, as described in the DoA. As such, this deliverable includes descriptions of the methodology and intermediary steps to reach conclusions on the interchange formats, including use case definitions, even though the title only describes the definition of the interchange formats as its topic.

In this deliverable, the third and final of three iterations, and an evolution of D6.1 and D6.4, we revise the second set of requirements for the prototype MeMAD platform. The aim of this single platform is to form a coherently integrated system of underlying technical components with a single interface for users to interact with, making it easier to test end-user workflows and to help assess the quality provided by various automated analytics and processing tools. The platform offers a single point of entry for audiovisual material ingestion, storage and workflow task dispatching, and provides a centralized metadata store and search index and interface.

This document refines the functional and non-functional (i.e., in terms of quality, processing performance or system resilience) requirements of the MeMAD prototype system, based on input concerning the tools developed in WP2, WP3, WP4 and WP5 and based on the project's use cases from which many requirements will be derived. In addition to (non-)functional requirements, this document adds specific test scenarios and evaluation criteria to determine the performance of the prototype system.

The structure of this deliverable is as follows.
We explain the methodology followed to obtain the MeMAD prototype requirements and exchange format specifications, and provide details on the context of use for the project, i.e., the relevant media production and consumption process. We list the overall project use cases and then for each, a set of more specific user stories. From this overview, we then deduct a sets of functional cases per topic which define the functional requirements for implementation as part of the second and final MeMAD platform prototypes. Additionally, requirements for each of the media and metadata processing components developed in other work packages in the MeMAD project are also provided. Finally, we describe a more concrete definition of the exchange formats that will be used between components of the prototype, taking into account the implementation requirements defined in the previous sections.

2 Changes with regards to deliverable D6.4

As mentioned, this deliverable is the third and final iteration of the D6.1 and D6.4 documents. With respect to D6.4, the following minor set of changes has been made:

1. The methodology described in Section 3 has been updated to reflect the current state of the specification and implementation plan for the MeMAD platform prototype, at Month 27 of the project.
2. The use cases and user stories have been revised to include final decisions on which functionality will be implemented as part of the project, and which stories will be discarded because they are effectively out of the scope of the project, because insufficient resources are available to implement them, or because their added value in terms of research value were deemed too low. In particular, user stories 1.3.1, 2.1.4, 2.2.3, 3.3.1, 3.4.1, 4.1.1, 4.1.2 and 4.1.3 have now been left out. The result is listed in Section 5.
3. The definition of the functional development epics has been refined where necessary to reflect new insights obtained from the second platform prototype implementation (cf. D6.5) and subsequent evaluation round (cf. D6.6).
4. As with Section 6, the requirements for each of the developed media and metadata processing components have been slightly revised in this final document, based on learnings from implementing the second platform iteration, as described in Section 7. One component was added for language segmentation and classification (cf. subsection 0).
5. Section 8, which concerns the actual data interchange format specifications has been extended to include definitions of formats that were previously still under development. In those few cases where the file formats are yet to be finalized during the course of the final project year, a close collaboration between WP6 and the other work packages has been set up such that the best possible interchange formats will be decided on. All format definitions are collected in an online Git repository such that an up-to-date collection of these specifications can always be obtained regardless of the state of this deliverable.

3 Methodology for determining prototype requirements and metadata exchange formats

This section describes the methodology followed for obtaining the MeMAD prototype requirements, of which the functional requirements and the data exchange format definition are a part. Our methodology is built on two pillars:

1. **As a guiding principle for defining the functional requirements of the project's prototype and its underlying individual components we use the four project use cases (PUCs) defined in the project's DoA.**
   The four PUCs define in broad terms the functional objectives of the project and give us a context to build more detailed functionality specifications from, even though they have been defined in a very generic fashion.
2. **For the definition of actual functional requirements, we follow the Human-centred Design methodology[2].**
   Applying human- or user-centered design (UCD) is a good match for the MeMAD project, as the project aims to build a prototype that will be actively used and interacted with by end users. Moreover, when using the prototype system these users will need to adapt to changes in the execution of contemporary production processes, because the MeMAD prototype will offer improved or new ways of tackling problems or it will deliver automated solutions of whose outputs need to be incorporated in existing production processes. Examples of such changes include: users who manually correct automatic suggestions for video clip descriptions instead of typing all descriptions manually, or users who are presented with automatically generated transcriptions of interviews in electronic format while they formerly only had this information available in paper print-outs.
   Using the UCD methodology will help the consortium build better user experiences because end users will be involved throughout the design and development, and additionally, designs will be iterated on and refined by user-centered evaluations.

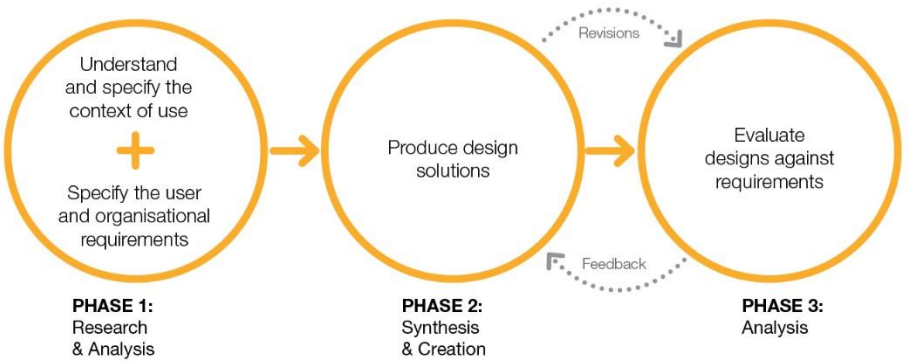

Figure 1: Synopsis of the User-Centered design process (from O'Grady, 2008[3]).

[2] Cf. ISO Standard 9241-210:2010 – Ergonomics of human-system interaction -- Part 210: Human-centred design for interactive systems.
[3] Cf. Visocky O'Grady, J. & Visocky O'Grady, K. (2008) The information design handbook. Mies: RotoVision.

3.1 Implementing User-centered design for MeMAD

The execution of UCD in MeMAD will occur in several steps, as illustrated by Figure 1.

1. Phase 1, step 1, is the research phase to understand and determine the context of use across the entirety of media production and consumption process, and from the viewpoint of the various envisaged stakeholders of the project's results. These stakeholders include:
    - End users who will be using the system first-hand.
    - Other stakeholders who have a stake in the implementation of the system, e.g., producers who manage the budget for the process execution.
    - Technology developers and researchers who need to understand the user requirements in order to implement them, and who need to provide feedback on the feasibility of succeeding in the implementation of user requirements.

    The context of use is further explored in Section 3.

2. In Phase 1, step 2, the actual functional requirements are defined. More detailed user requirement will be defined as user stories to describe more specific sets of desired functionalities, each of them fitting within the definition of one of the PUCs on the one hand, and with the context of use we determine from step 1. We use user stories for this purpose, from an end-user perspective (implying desired functionality from the back-end system indirectly) because they are easy to grasp by project stakeholders, which will facilitate their evaluation. At the same time, they form a good basis to refine further requirements from.
   To ensure the user stories we define are relevant to all project stakeholders, we will validate them in a project review process as follows:
    a. To make sure we cover the entire spectrum of possible applications for this project and the project's use cases, we will consider the media production and consumption process end-to-end (as introduced in Section 3) to determine possible innovative functionalities that the MeMAD project can provide. To give us a first baseline to start from and allow for easier discussions, the consortium partners define this first set of stories, aided by industry media professionals employed by consortium partner YLE.
    b. While YLE's representatives count as experts in their domain, to avoid undesired bias by including a single organisation's viewpoints, the candidate user stories will also be evaluated by members of the project's external collaborators and by contacting professionals through industry channels such as EBU and the EU Mediaroad sandbox project.
       These stakeholders provided with a survey in which they could indicate their interest in each user story, along with the possibility to suggest additional scenarios for implementation by MeMAD. New and updated user stories were added to Section 5 of D6.4. For this deliverable, a final decision has been made regarding the actual implementation of each user story by the project consortium, and we provide a justification for excluding each discarded user story.

3. In Phase 2, based on the finalized list of relevant user stories, the exact requirements involved for each story are further refined. The result from this effort is the following:

a. Detailed requirements that guide the development of the prototype and underlying technology components by defining the functionality required from the system to realize the goals stated. This encompasses textual descriptions at a first stage and will be supplemented with visual designs and potentially interactive mock-ups of application interfaces. This process will be supported by in-depth interviews and interactive design sessions with relevant end users, based on contemporary production processes and tools determined in Phase 1.
   In addition, non-functional requirements will also be deduced from the context of the use (e.g., timing or processing speed constraints, availability constraints, accuracy requirements, etc.) along with criteria to measure the success of each non-functional requirement.
   The new Section 6 of this deliverable provides the first definition of these required functionalities, grouped in a set of 10 implementation 'epics', each of which implements one or more of the project user stories.
b. To the definition of the functional epics we have also added the requirements for each underlying processing component – operating either directly in the visual and auditory domain, and/or using content metadata as a starting point – that is to be delivered for integration with the prototype platform, in Section 7. The requirements of these 15 components are derived from the definition of the functional case from Section 6.
c. Finally, Section 8 defines more concrete specifications of the file formats that will be used for exchange of data between the project's processing components (as defined in Section 7).

4. Finally, for Phase 3, exact test cases and evaluation procedures will be defined to measure the implementation of prototype features against the functional and non-functional requirements defined in Phase 2.
   This report already contains a tentative set of evaluation criteria to help the consortium members gain a better understanding of what will be expected from their implementations and how their components will be tested. These evaluation criteria will be extended in the preparation for the second and final evaluation rounds of the prototype (as reported on in D6.6 and later on in D6.9).

5. Once an initial cycle has been completed from Phase 1 to Phase 3, further iterations between Phase 2 and Phase 3 will occur to first tweak the functional requirements of the systems implemented in the project, based on user-centered feedback, and secondly to build an improved implementation, which will then again be evaluated. This is illustrated in Figure 1.

As explained above, because the UCD process encompasses the entire duration of the project, not all of its results are available in this deliverable yet. We summarize how each piece will subsequently be completed in which deliverable of Work Package 6 in Table 1.

Interchange format specification and requirements definition	The MeMAD prototype	Evaluation of the MeMAD prototype
D6.1: Definition of the context of use and an initial set of high-level user requirements. In addition, this deliverable maps out a first revision of required metadata and sets the requirements for the first prototype iteration (M3).	**D6.2:** A report on the first implementation of the prototype, executed per the specifications of D6.1 (M12).	**D6.3:** An evaluation of the first prototype and its requirements, to the extent possible with the limited implementation. This report also includes feedback concerning the use cases and requirements for exchange format specifications (M12).
D6.4: Refinements of the initial set of high-level user requirements based on feedback from external advisors. This second version will define more detailed requirements for the second MeMAD prototype, including test criteria and scenarios (M18).	**D6.5:** A report on the implementation of the second prototype, executed per the specifications of D6.4 (M24).	**D6.6:** An evaluation of the second prototype and its requirements (M24).
D6.7: Definition of the final requirements and test criteria for the MeMAD project prototype, along with final specifications of all metadata exchange formats (M27).	**D6.8:** A report on the implementation of the final MeMAD prototype, executed per the specifications of D6.7 (M36).	**D6.9:** A report on the evaluation of the final MeMAD prototype, which will be done by interested parties outside the project consortium (M36).

Table 1: Orientation of MeMAD Work Package 6 deliverables.

4 Processes and stakeholders in the media production and consumption chain

To better understand the context of use for MeMAD technologies, we need to take a closer look at the media production chain, and identify each of its processes, along with the users and other stakeholders who participate in these processes and for which implementing MeMAD applications could make sense.

Two large process phases are relevant for the MeMAD project: the production and consumption of media. The consumption phase can be considered by itself as a single process executed by the consumer end user, who can be a viewer, listener, reader or a combination depending on how the media is delivered.

The production phase on the other hand encompasses many processes that we need to map out in detail to understand the MeMAD context of use. The production chain can be roughly divided in the following sequential phases:

1. **Pre-production and conceptualization**, which involves the phase of story building, conceptualization of programs and the planning of the further production processes;
2. **Production**, which involves the production of original audiovisual material;
3. **Post-production**, which involves the assembly and finishing of various pieces of audiovisual material, which is either originally produced in phase (2) or reused from existing sources or archives;
4. **Distribution**, which involves the preparation of the distribution of audiovisual content, including taking care of accessibility and delivering programs in certified formats to distribution outlets.

The pre-production and conceptualization processes are out of scope for MeMAD. We hope to re-use some of the results that are produced at this stage, e.g., production scripts, but as no audiovisual content exist at this point, it is not of particular interest to this project.
Looking further at the production phase, in which new content is being recorded on-site, on-set or in studios, there is also limited benefit to be obtained from the MeMAD project: the acquisition process for new content is performed based on input from pre-production and is executed using highly optimized and specific equipment and procedures. These can be influenced by feedback from the post-production process (e.g., a crew needs to shoot another piece of material because a particular viewpoint was still missing when assembling the program) but it would not inherently be improved by MeMAD tools (as defined in the PUCs).

The interest for MeMAD lies with the post-production and distribution processes, as listed in Table 2.

Production Phase	Task/Process	Users and stakeholders
Post-production	**Material collection:** users gather interesting audiovisual content to build programs from. This content can come from original acquisition (obtained from the production phase) or from archives. Documentaries often source much content from archives, while current affairs programs make combinations of both sources (e.g., newly recorded interviews mixed with archive content), and drama production is often exclusively comprised of newly produced material. Users will find material guided by conceptual outlines or stories that were made during pre-production.	**Documentary/current affairs production team, news reporters, journalists,** share similar roles (depending on the specific program format they work at), which is to gather new materials (they might actually be responsible for production part of the material itself) and assemble them according to the program concept. They will also assist editors in making the best editing decisions (cf. Editing process). **Producers**, will oversee the production of programs or items and will guide the creation process towards a desired outcome. **Researchers**, will gather materials from archives based on research done on a specific topic.
	Editing: this process involves the fine-grained assembly of the program by taking individual pieces from the gathered materials and 'cutting' them into a montage using interleaving pieces of content. Editing can happen for video and audio separately, or combined, depending on the program format.	**News editors, Documentary editors, Archive editors, Drama editors, sound editors**, depending on the program format, each editor has a similar function, but with particular expertise for best delivering a certain format to the screen, will cut and assembly various pieces of audiovisual content into a final presentation. **Directors,** who possess the creative control over the production will assist the editor in making the correct editing decisions.
	Production office: which involves a variety of tasks, including resource scheduling and planning, keeping track of progress of the overall production process, and ensure the program is delivered on time and budget. This means that the production office oversees all other processes in post-production and delivery, and is hence a stakeholder in those processes too.	**Producers**, who in this role supervise the budget of the production and ensure that the program is delivered for the smallest budget possible, while enabling creative visions to be expressed wherever possible.
Distribution	**Subtitling:** which involves the creation of textual subtitles or	**Subtitlers**, who create subtitles for programs that will be

Distribution (continued)	closed captions to help audiences understand the program's content.	broadcast. This can involve creating same-language subtitles for accessibility purposes, or translated subtitles for enabling the material's access in a given language market. Depending on the subtitling context, the procedures and tools used will be different: live subtitling is done in real-time, often using re-speaking ASR technologies, while off-line subtitling is done in batch using dedicated subtitling tools.
	Audio description, which involves the creation of auditory descriptions of the content depicted in the program, often interleaved with original content audio, e.g., character dialogue or sound effects.	**Audio describers**, who create the audio descriptions, first by writing a script, then by resolving timing such that the descriptions properly interleave the original audio content, and finally by voicing, recording and assembling the script into a final audio-described mix.
	Archiving, which deals with managing material coming into archives and ensuring content is placed in the archive such that it can be retrieved as efficiently as possible.	**Archivists,** who curate the metadata that is input into the archive to ensure all content is annotated in a uniform fashion to ensure maximum retrievability of archived content.
	Delivery, which deals with the logistics of delivering programs (typically as digital files nowadays) to distribution chains, broadcasters, OTT services, etc. Alternatively, it also deals with delivering content to consumers as efficiently as possible (e.g., by maximizing the content that is being consumed, or by maximizing revenue by promoting content which delivers higher monetary returns or drives better received advertisements).	**Production officers**, who deal with material logistics when finishing and delivering programs, in the right format and with the proper metadata associated (e.g., program identifiers, order numbers, etc.). **Marketing executives**, who deal with optimizing the revenues vs. costs of the delivery services, and who want to promote as much of their service's content as possibly while maximizing revenue, e.g., by enabling content-related advertisements.

Table 2: Processes and stakeholders in the media production and consumption chain.

5 Use cases and user stories for an integrated MeMAD prototype system

In order to determine the functional requirements for an integrated MeMAD prototype system, the consortium members have refined the original four project use cases into an extensive set of candidate user stories which describe the actual potential functionality required from the MeMAD system in more detail. We have done this by intertwining the PUCs with the processes (and stakeholders) identified in the previous section.

In this final revision of this requirements document, we provide a final assessment of those user stories that will be implemented in the remainder of the MeMAD project. For each of the discarded functionalities, we provide a justification of why that piece was left behind, and we indicate that leaving out these user stories does not impact the project goals. The decisions regarding which functionalities to discard were made over the course of two plenary project meetings (at University of Surrey in September 2019 and at INA in February 2020), and a number of remote consortium teleconferences on this topic over the course of 2019.

The following set of tables describe the user stories and the resulting (high-level) functional requirements, along with the relevant users for whom the functionality is provided.

5.1 Project Use Case 1: "Content delivery services for the re-use by end-users/clients through media indexing and video description"

Online media delivery platforms rely heavily on media metadata in supplying, recommending and grouping digital media to clients. This use case aims to enhance the end-user experience of such services by creating and making use of rich metadata and hyperlinking through the use of automated media analysis and multimodal media indexing.

As a result, users of such delivery services will be able to discover and watch media that are meaningful to them from a spectrum of starting points and interests that is significantly broader than what can be achieved by current methods of metadata creation. Users will, for example, be able to browse and discover themes, people and places from media, and parts of media containing these even when the information has not been entered by production staff or when the original media product was designed for a different purpose.

With respect to the media production process, this use case focuses on the consumption process, when actual production has completed. As such, we consider only requirements that deal with content consumer end users.

5.1.1 Sub-Use Case 1.1: The user can discover media content about a specific theme, person, place.

Considering entire programs, this sub-use case deals with how end users can discover related content through a variety of dimensions of metadata that is associated with the media content.

User Story	Description	Users
1.1.1 – Searching for consumer content.	Users can search directly for content thanks to metadata associated with all consumable content. The associated metadata exists across several dimensions and topics, incl.: persons, locations, time periods, subjects, etc. This way, users can, for example, locate content dealing with a specific topic such as furniture design, German politics, 1920's lifestyle, cycling, etc.	Consumers
1.1.2 – Finding related content.	Users can discover related content thanks to metadata enrichments added to consumed content. Properly distinguished relations can further refine the accuracy of these relations. Examples include: a user is interested in other content related to the current by way of a place of living or a time period, or a user is interested in related content because it shares the presence of relatives or prominent figures.	Consumers

Table 3: User stories for sub-use case 1.1.

5.1.2 Sub Use Case 1.2: Getting the relevant parts from the program.

Not only entire programs can be cross-related and searched for, but also parts of a program. By relating program parts, searches can become more accurate and users can be provided with even more flexibility in consuming relevant content.

User Story	Description	Users
1.2.1 – Finding related program segments.	Users can access individual program segments, instead of accessing entire programs through which they then have to filter the relevant sections manually. Examples of this story are the following: • In a lifestyle or current affairs programme, users can find and retrieve those segments which are of interest to them, e.g., dealing with a specific topic, discussing people of interest, etc. • Users can find all quotes on a certain topic pronounced by a public figure and be able to listen and to see them.	Consumers
1.2.2 – Skipping program segments.	Users can skip those segments from a program that are not of interest to them. This could include also skipping the end credits and opening graphics automatically between episodes.	Consumers
1.2.3 – Consumer content is auto-segmented into relevant segments. (Added in D6.4)	To make content more accessible, users are presented with segments that are automatically deduced from larger programs. These segments are thematically delineated from each-other, and show summarizing associated metadata, e.g., the relevant topics or persons in this segment. This functionality allows users to better find smaller relevant parts of content within larger opaque programs.	Consumers
1.2.4 – Users can use a visual search feature to help them find previously un-annotated content. (Added in D6.4)	This visual search feature helps the users to find undescribed material (faces, objects, scenes) that they find interesting. To this end, users can easily train new classifiers implicitly by example with the aim of performing a visual search on the topic of their interest.	Consumers

Table 4: User stories for sub-use case 1.2.

5.1.3 Sub Use Case 1.3: Gaining insights into program consumption.

Providing executives with statistics that relate user consumption behavior and program topics (obtained through its associated metadata) will provide them with better insights into which (combinations of) topics perform well with consumers. Additionally, trend analysis will allow them to learn evolutions of topic popularity based on changes in consumption of content related to specific topics.

User Story	Description	Users
~~1.3.1 – Tracking content consumption in terms of associated content metadata.~~ ~~(Added in D6.4)~~ **Discarded in D6.7:** While this user story was added based on select ECG member feedback, its implementation was deemed too derivative and considered outside the scope of the project, which wishes to focus foremost on the creative and efficiency aspect of content creation and disclosure processes.	~~Users can analyze consumption patterns of audiovisual content based on a combination of content metadata and consumer behaviors. By including grouping and summarization of associated metadata for use in analytics about media consumption interesting conclusions can be made about the relationship between an items topics and its consumption: e.g., to analyze content "play starts" per character or named topics identified in the content.~~	~~Marketing executives.~~

Table 5: User stories for sub-use case 1.3.

5.2 Project Use Case 2: "Creation, use, re-use and re-purposing of new footage and archived content in digital media production through media indexing and video description"

This use case aims to improve discoverability and re-usability of digital-born as well as pre-existing media for the purpose of crafting new stories and audiovisual concepts. Media professionals are provided with rich and relevant relationships between archive media, scripts and raw footage during different stages of digital media production, enabling them to develop a digital story and concepts with the help of automated metadata extraction and media analysis. Relevant media fragments are automatically recommended, which saves significant amounts of editorial work compared with conventional methods of research in media archives.

With respect to the media production process, this use case focuses exclusively on the actual creation process, which for our project begins from the moment audiovisual content is created or recuperated and the assembly process can start, right up to finishing content for delivery. The focus of the requirements hence lies with the professional media producers.

5.2.1 Sub Use Case 2.1: Ingest, organization and editing of new footage.

After the very first stages of the media production process where the program is conceptualized and its story elements are defined, the first opportunity for MeMAD to provide meaningful added value is presented: newly created material enters the production facility at an initial stage, and it can then be used for editing and shaping the story into an actual program. This is the subject of requirements for this sub-use case.

User Story	Description	Users
2.1.1 - Real-time analysis and indexing of ingested content.	Reporters return from the field with interviews and other footage. They ingest the material into the production system which indexes the files with rich metadata. The indexed data offers quickly several alternatives for interviews and footage to be used in a very short time span, to ensure the resulting program can be completed the same day.	News and current affairs reporters.
2.1.2 – Extensive analysis of ingested content.	Documentary production teams return with a large collection of raw footage, which they ingest into the production system. The system indexes the files so that the production team can move on with scripting and editing their program. Typically, the amount of media is quite large, but the production schedule is not as tight as on day-to-day news production.	Documentary and current affairs producers.
2.1.3 – Users browse ingested content for editing.	News editors go through news feed material without pre-existing knowledge about the content and choose an interesting topic to edit a news story on. Instead of starting from a given set of search terms of topics, the ingest library could offer a list of random or	News editors, documentary makers, journalists.

(2.1.3 – continued)	popular topics for which content has recently been ingested and processed, to kick-start the content discovery process.	
~~2.1.4 – Ingest feed notifications.~~ **Discarded in D6.7:** The implementation of this user story was evaluated by the consortium but was found to lack innovative potential and its added value was considered low compared to e.g., spending effort on the implementation of other added user stories such as 2.2.6 and 2.2.7.	~~News feeds are constantly monitored and analysed as they feed into the production system. Real-time processing provides speech recognition and keyword spotting, allowing for trend analysis of the detected results. Thanks to such analysis, incoming feed topics can be tracked and potentially newsworthy content can be detected.~~	~~News editors, journalists.~~
2.1.5 - Editing assistance using multi-model metadata.	Editors can take advantage of multimodal annotations of content to help speed up the editing process by quickly triaging material before editing. Examples include: • Occurrences of detected persons in the image are indicated on the editing timeline; • Transcript annotations are available on the editing timeline; • Automatic classification of shot types (close-up, two-shot, over-the-shoulder).	All editors, incl. news editors, documentary editors, current affairs editors, etc.
2.1.6 – Use of autotranslated content for editing.	Editors who are editing interviews conducted in a foreign language unknown to them can get to work immediately, without the need for any available interpreters because the content has been automatically transcribed and machine-translated.	All editors, incl. news editors, documentary editors, current affairs editors, etc.

Table 6: User stories for sub-use case 2.1.

5.2.2 Sub Use Case 2.2: Discoverability of archive content.

Not all content used for creating audiovisual programs is newly created for that single program. Often, material is reused from archives, where the challenge is to disclose as much relevant content from these archives. This is not a trivial task, as contemporary processes can only rely on manually entered metadata for searching. MeMAD can help resolve this issue by retro-actively processing and enriching archived content such that it becomes easily discoverable for re-use in new productions.

User Story	Description	Users
2.2.1 – Searching for content in archives. (Revised in D6.4)	When searching through the archive, users can find material using metadata that has been added automatically, and optionally been corrected by archivists. As with user story 1.1.1, this associated metadata exists across several dimensions and topics. Researchers can browse using detected topics, persons, speech fragments, music and image characteristics, detected emotions, etc. using named entities or free text queries. Examples include: • Looking for content about a given celebrity who has recently deceased; • Looking for footage of an Airbus A380 taking off from Charles de Gaulle on a foggy morning; • Looking for specific quotes uttered by a politician who was in the news yesterday. • Looking for close-up shots of a given person.	Researchers, journalists, editors.
2.2.2 - Searching for segments of content in archives.	As an extension to 2.2.1, researchers can also retrieve just those sections that are relevant to the search query of the user. E.g., news editors want to find the correct one sentence quote from the video recordings of parliamentary meetings.	Researchers, journalists, editors.

~~2.2.3. Notifications from the archive about selected topics.~~ **Discarded in D6.7:** As with user story 2.1.4, the implementation of this user story was evaluated by the consortium but was found to lack innovative potential and its added value was considered low compared to e.g., spending effort op the implementation of other added user stories such as 2.2.6 and 2.2.7.	~~Researchers can set up notifications such that they are alerted to new content in that matches their search criteria. E.g., a news editor instructs the system to watch content from the city council meeting and to identify potentially interesting content pieces.~~	~~Researchers, journalists, editors.~~
2.2.4 – Intuitive manual correction of automatically generated metadata. (Revised in D6.4)	Archivists can correct automatically tagged and enriched archival items. This must be done using an intuitive user interface such that this process will take much less time than inputting all metadata manually. Additionally, these manual corrections could also include modifications to facial or speaker recognition profiles such that future detections occur with greater efficiency. Finally, provisions should be made to ensure automatically added tags can be clearly distinguished from those validated by a trusted source.	Archivists.
2.2.5 – When looking up archival content, hyperlinked related media are also shown. (Added in D6.4)	Just as consumers, researchers and journalists can be helped in their search for content when provided with background information – and even relevant links - on topics that are addressed in archive content. Examples are similar to those described in user story 3.2.1.	Researchers, journalists, editors.

2.2.6 – Users are presented with auto-summarized segments of archival content. (Added in D6.4)	Different users and use cases benefit from different amount of detail in the data they use. For example, a full transcript (for journalists) vs. a few keywords for summarizing the main topic of an entire interview (for researchers). Similar to user story 1.2.3, to make content more accessible, researchers and journalists are presented with segments that are automatically deduced from larger programs stored in the archive. These segments are thematically delineated from each-other, and show summarizing associated metadata, e.g., the relevant topics or persons in this segment. This functionality allows users to better find smaller relevant parts of content within larger opaque programs.	Researchers, journalists, editors.
2.2.7 – Users are suggested auto-generated stories from archive content that they can modify and shape into final program items. (Added in D6.4)	Researchers and editors can create a search query that results in an automatic story constructed from relevant and related archive content segments. They receive a story proposal where they can efficiently change the selection and the order of the footage.	Researchers, journalists, editors.

Table 7: User stories for sub-use case 2.2.

5.2.3 Sub Use Case 2.3: Managing material and footage between multiple production parties

After an audiovisual program is completed there remain a variety of opportunities for the MeMAD project to help facilitate in helping with exchanges of material, and their associated metadata, between different production parties, e.g., between the production house and the broadcaster, or between the production and an archive.

User Story	Description	Users
2.3.1 – Tracking media assets in final programs.	Archive researchers look up promising video clips for a TV production and deliver them to a production house responsible for the production. Later on, the production house wants to track which segments from which archive clips were used in the finished program.	Researchers, producers, rights managers.
2.3.2 – Delivering relevant production metadata downstream.	After finishing a joint production, a production company delivers the finished TV series to a media archive and the production officers sending the files needs to add content description and metadata to them based on the guidelines from the receiving archive.	Production officers, archivists.
2.3.3 – Processing and harmonizing delivered production metadata.	Archivists at a media archive receive finished TV programs from multiple production companies. Some programs may have partial metadata or content descriptions, but archivists need to produce coherent metadata for all to enable consistent further archive use.	Archivists.

Table 8: User stories for sub-use case 2.3.

5.3 Project Use Case 3: "Improving user experience with media enrichment by linking to external resources."

A video program may be edited using a complex narrative, but viewers have different background and interests and may not be familiar with all the elements being presented, triggering the need to go more in depth for some aspects being presented. Video programs also trigger social media reactions (e.g. on Twitter or Facebook) where sometimes viewers clip and repurpose some original parts of the video program. One way to improve the user experience is to provide individual users the possibility to access and explore related material (e.g. videos, news articles or set of facts extracted from encyclopedia) that will contain additional information that they personally need or are interested in to better understand the narrative of the video program.

External material may be essential for understanding the audiovisual content. For example, when republishing decades old audiovisual content from the archives, to understand the meaning of the archive content, additional material may be required that gives the historical context and information on how to interpret the content.

With respect to the media production process, as with use case #1, this use case focuses on the consumption process, when actual production has completed. As such, we consider only requirements that deal with content consumer end users. To the extent that the user stories defined here require metadata to be made available during the media creation process, they will have a counterpart user story from use case #2 or #4.

5.3.1 Sub Use Case 3.1: Promoting relevant cross-platform media content.

In this sub-use case, a variety of related content, both linearly audiovisual content but also interactive and cross-platform experiences is recommended to users as part of navigating content libraries.

User Story	Description	Users
3.1.1 – Libraries of Audiovisual content hyperlink to various related media during browsing.	Users browsing on-demand over-the-top (OTT) services can select interesting topics or headlines, which refers them to audio and video clips related to the first broadcast and textual content describing how the different media clips are related to the topic, in addition to containing references to news articles that were produced about this topic.	Consumers.

Table 9: User stories for sub-use case 3.1.

5.3.2 Sub Use Case 3.2: Extending the user-experience with more details and background information about the content.

Providing users watching audiovisual content with a rich-media experience of linked content that relates to the consumed content can provide valuable insights for those users. Additionally, it could also lead additional discovery of existing but seldomly accessed rich media content.

User Story	Description	Users
3.2.1 – During playback of Audiovisual content hyperlink to various related media are shown.	Users watching documentary or current affairs content through an on-demand service are provided with background information – and even relevant links - on topics that are addressed in the program. For example: • For users watching a program about animals in Sahara, an on-demand service displays information about the currently visible objects, such as ants, birds and plants; • Users listening to radio programs about birds are presented with information about the birds being discussed on a 'second' screen. • Users watching current affairs programs in which politicians are features are presented with linked content to clarify each politician's background and affiliation, along with party programme points that this politicians party stands for.	Consumers
3.2.2 – During playback of Audiovisual content hyperlink to various interactive media are presented.	We provide three examples to sketch possible scenarios for this user story: • Users watching sports content through an on-demand service are provided with statistics about the players and game, along with relevant historical statistics and links to further information. • Users watching lifestyle programs through an on-demand service can participate in discussions with like-minded consumers related to the topics addressed in the program. • Users watching a health program about diabetes are also shown an interactive experience, "are you in risk of getting diabetes?" to test their own risk of obtaining the disease.	Consumers

Table 10: User stories for sub-use case 3.2.

5.3.3 Sub Use Case 3.3: Validating the content, e.g. the truthfulness.

Providing users with links to related rich media content without curation can present hazards, as the linked content might not always present truthful and accurate information. At the same time, the original content might suffer from the same issues. We can envision a potential role for the MeMAD project in enabling insights into the truthfulness of the content that is consumed and linked to.

User Story	Description	Users
~~3.3.1 – Truthfulness validation of audiovisual content.~~ **Discarded in D6.7:** While this user story describes functionality with great and crucial merit, it is not within scope of the project, and not sufficient resources could be spent on a proper implementation. This kind of functionality could be integrated into the platform later when obtained from a service built by other projects, e.g. the EU H2020 Fandango[4] and WeVerify[5] projects.	~~Users watching news, current affairs or political programs are presented with results from a truthfulness analysis based on the content's analysed speech and externally linked resources, giving an indication whether what is being said on screen is plausible to represent the truth, or is likely fake news.~~	~~Consumers.~~

Table 11: User stories for sub-use case 3.3.

5.3.4 Sub Use Case 3.4: Show relevant TV or other advertisement in context of the current content.

Content providers can benefit from targeted advertising, which is related to the content being distributed, because it is more relevant to consumers than generic advertising. Regardless whether the user's personal preferences are taken into account, or the advertising is based only on the profile of the content, MeMAD-generated and managed metadata can assist in advertisement recommendations.

User Story	Description	Users
~~3.4.1 - Content-related advertisements.~~	~~OTT distribution services send out targeted content-related advertisements. Instead of~~	~~Consumers, OTT distribution producers.~~

[4] Cf. The Fandango project website at: https://fandango-project.eu/.
[5] Cf. The WeVerify project website at: https://weverify.eu/.

Discarded in D6.7: For the same reason that user story 1.3.1 was left out, this user story is also discarded. The focus of the project is on the content creation process and on how this directly reflects on the end-user content consumption experience. As such, we decided to drop this user story.	~~showing generic commercials, the OTT service can benefit from associated media item metadata to show advertisements that are likely more relevant and of interest to viewers.~~ ~~At the same time, the OTT service can sell this advertisement space in a targeted way, e.g., bicycle manufacturers can bid on advertisement slots associated with sporting events or cycling documentaries.~~	

Table 12: User stories for sub-use case 3.4.

5.4 Project Use Case 4: "Automated subtitling/captioning and audiovisual content description. Speech and sounds to text and also visual content to text, both with multiple output languages, for general purpose use and for the deaf, hard-of-hearing, blind, and partially-sighted audiences."

This use case addresses an urgent requirement to enhance as much content as possible with complementary subtitles and verbal or aural content descriptions. Conventionally these are created by human subtitlers, audio describers and translators, and at a total production cost of 1000-1200 Euro per hour (for subtitling) up to 3000 Euro per hour (for audio description). Also, manual subtitling and audio description requires a significant cycle time from one to two weeks. For this use case, we will undertake to maximize productivity of both subtitling (same language as well as language to language) and audio description processes, through "supervised automation".

This is the single PUC which is clearly represented both in the production and consumption process. The 'consumption' of subtitling and (audio) content description, especially if targeted toward minority groups of audiences for accessibility purposes, should in particular have a consumer counterpart such that the project can properly take into account the consumption environment and the consumer quality requirements that will be posed on any generated subtitles or content descriptions. Meanwhile, of course, the actual production processes involved in making these elements are an important focus for MeMAD.

5.4.1 Sub Use Case 4.1: Live / near-live captioning, subtitling and audio description.

MeMAD has the potential to assist in optimizing contemporary accessibility production processes such as same-language and intralingual subtitling and audio description. In this first sub use case, we consider the processes that already exist today and that could

be helped by the MeMAD components, without profoundly impacting common production practices. The implementation of these use cases would, however, require further research into their actual design, feasibility and effectiveness.

User Story	Description	Users
~~4.1.1 – Assistance in live subtitling.~~ **Discarded in D6.7.** See below.	~~Subtitlers who are live subtitling (i.e., with a minimal delay wrt. the broadcasted program, measured in seconds) could be aided live ASR results that provide suggested subtitles which only need correction.~~	~~Subtitlers.~~
~~4.1.2 -Assistance in live audio description.~~ **Discarded in D6.7.** See below.	~~Similarly, audio describers who are describing live broadcasts to aid visually impaired people could be helped with suggested automated descriptions of the content (e.g., automatic identification of people in the image).~~	~~Audio description producers.~~
~~4.1.3 – Assistance in near-live subtitling.~~ **Discarded in D6.7.** See below.	~~Near-live situations create less time pressure to deliver subtitles than live scenarios have different dynamic and allow MeMAD tools to help in this process. The suitability compared to live subtitling should be investigated in this case.~~	~~Subtitlers.~~
4.1.4 – Automated same-language subtitling.	Users, and in particular, hearing-impaired users are provided with automatically generated same-language subtitles for content such that they can consume the content without the audio being available or audible.	Consumers, Subtitlers.

Table 13: User stories for sub-use case 4.1.

The project consortium has decided to discard the user stories with functionality that operates in a live material processing context. Given that the dynamic of a live production context is different from one that processes material in bulk (the volume is well-known and dimensioned but needs to be processed in real-time and at a guaranteed level of quality), these processes are often highly optimized within their own scope. For example, live ASR re-speaking technology (trained specifically for the materials being processed and for the respeaking voice) is often employed to help realize real-time subtitling. Given the state of the MeMAD subtitling software implementation observed during the 2019 evaluations (cf. D6.6), we concluded that more effort needs to be spent

on improving the ASR and subtitle generation service in non-live settings first, before trialing them in a wider context. As such, these user stories are left out of scope for the remainder of the project. The final evaluation on subtitling will however point out if and where gains are to be made when MeMAD technologies are adapted for live production scenarios. These conclusions will be reported in D6.9.

5.4.2 Sub Use Case 4.2: Extending coverage of audio descriptions

Whilst the automation of live and near-live audio description as outlined above is likely to be a longer-term goal, finding ways to extend the coverage of audio descriptions produced off-line, without proportionally increasing the effort to create these descriptions for more content is an important aspect of the MeMAD project. At the same time, it is a very challenging one: auto-generating audio descriptions which correctly capture the semantics of the audiovisual content and transcend the level of plainly describing what is visible and audible to take into account the editorial context of the content will be non-trivial to prototype.

User Story	Description	Users
4.2.1 – Content consumption with auto-generated audio descriptions.	Visually impaired consumers can experience all episodes of their favorite shows, thanks to the audio descriptions that have been made available using an additional audio track.	(Visually impaired) consumers
4.2.2 – Manual corrections improve auto-generated audio descriptions.	Audio description producers in charge of delivering audio descriptions can deliver content descriptions more efficiently thanks to automatically generated audio descriptions, that are reviewed and corrected manually.	Audio description producers.

Table 14: User stories for sub-use case 4.2.

5.4.3 Sub Use Case 4.3: Automatic translation of existing subtitles to other languages to increase minority or general audience accessibility.

The availability of subtitles associated with audiovisual content is often the most straightforward way of lowering barriers towards new audiences: textual subtitles can be delivered via side-channels and provide meaning to any foreign-language content. Making additional subtitles in other languages available to new audiences at marginal cost is an important topic for the MeMAD project.

User Story	Description	Users
4.3.1 – Automatically translated subtitles for foreign users.	Users with different language needs abroad can select the automatically generated subtitling in a language familiar to them such that they can follow the program's content.	Consumers.
4.3.2 – Automatically translated subtitling of foreign content.	Users browsing foreign European media libraries can consume this content even if it is produced in other languages. Thanks to automatically translated subtitles or audio descriptions, users can experience and understand content otherwise inaccessible to them.	Consumers.
4.3.3 - Translated subtitles based on translated transcripts	Subtitlers can create translated subtitles using translated transcripts as a starting point.	Subtitlers.
4.3.4 - Manual correction of auto-translated subtitles.	Subtitlers need to manually correct automatically translated subtitles because the automated translation will generate errors, and subtitle timing or wording sometimes need to be changed to deliver subtitles of sufficient quality.	Subtitlers.

Table 15: User stories for sub-use case 4.3.

5.5 Impact assessment of discarded user stories

To ensure the project's objectives are not impacted by the discarded user stories, we devised the following impact assessment, structured according to the four original objectives stated in the DoA.

5.5.1 Concerning O1: Develop novel methods and tools for digital storytelling

Even though user stories 2.1.4, 2.2.3 (about various content notifications) and 4.1.1, 4.1.2 and 4.1.3 (about live subtitling and audio description) have a potential impact on the development of methods and tools for digital storytelling, this impact is not significant to disrupt the execution of O1. Adding live processing scenarios can be viewed as an extension of the techniques built in the prototype, and the basis for novel methods for storytelling remain in place even without live processing. In particular, working with automatically enriched and machine-translated metadata for subtitling, content retrieval and auto-generation of stories deliver the novel storytelling methods aimed for by the first project objective. The same holds for the user stories dealing with notifications, which would represent a nice-to-have addition to optimize an end user's workflow, but they would not represent a fundamentally different way of content creation.

5.5.2 Concerning O2: Deliver methods and tools to expand the size of media audiences

The primary means of expanding media audience sizes in MeMAD is to enable access to media either by making it discoverable from large volumes of opaque content, or by making it understandable by new audiences, e.g., through translation or by providing a service that helps overcome physical impairments.

Performing truthfulness analysis (user story 3.3.1) would have been an addition in obtaining larger audiences by building additional audience confidences. However, as limited resources are available during this project it could not be incorporated into this project properly. Other projects feature this topic more prominently (as mentioned above).
Similarly, targeted advertisements (user story 3.4.1) would provide another tangential but indirect way of enlarging audiences by increasing the (possible) commercial adoption of the MeMAD technologies leading to an increased audience in this way. However, in terms of research, the implementation of this functionality was deemed too applied and somewhat far-fetched compared to more concrete challenges faced by this consortium. Because the audience expansion through advertising would be a second-order effect not due to enrichments of the content itself, we prefer to drop it out of scope and focus foremost on the other applications.

Finally, the live media production scenarios would (user stories 4.1.1, 4.1.2 and 4.1.3) certainly serve to expand media audiences by opening live content with better or more efficient subtitling and (audio) content descriptions. As such, this was the hardest scenario to trade off with regards to implementation. Ultimately, it was decided to prioritize the development of the foundation of the content enrichment scenarios and

underlying services in offline batch processing scenarios. When successful, these foundations can then be extended and optimized for supporting live media production use cases, but this will take place outside the work plan of this project.

5.5.3 Concerning O3: Develop an improved scientific understanding of multimodal and multilingual media content analysis, linking and consumption

Even without the discarded user stories, sufficient applications of multimodal content analysis and machine translation of metadata remain to help the consortium develop the sought after understanding of these multimodalities. In particular, the interlingual subtitling, video captioning and automated description, and media memorability detection are pertinent applications. The analysis of truthfulness would have added a nice-to-have dimension to this research. However, insufficient capacity is available in the project to properly deal with this topic while other projects exist that solely focus on this research area.

5.5.4 Concerning O4: Deliver object models and formal languages, distribution protocols and display tools for enriched audiovisual data

Even taking the discarded user stories into account, all defined processing components (cf. Section 7) remain required for the execution of the MeMAD platform. As such, the requirements and delivered implementations of object models, distribution formats and protocols and visualization tools for audiovisual enrichments are not impacted.

6 Functional epics of implementation

In order to make the list of use cases and user stories more insightful and manageable with regards to the implementation to be performed, we have grouped them into 10 functionally related epics. These epics combine the implementation work described by one or more user stories which share a single objective or that represent similar required functionality[6]. For each epic we can more easily define requirements and evaluation criteria such that unnecessary duplication of efforts is avoided. The following epics have been defined:

6.1 Searching for and locating related consumer content

6.2 Auto-enrichment of ingested and archived content

6.3 Searching and browsing for ingested and archived content

~~6.4 Notifications of available ingested and archived content~~

6.5 Editing assistance using multi-modal and multi-lingual metadata

6.6 Auto-generation of stories from archived or ingested content

6.7 Delivering and processing finished program metadata

6.8 Semantic enrichment of content and linking of external resources

6.9 Searching for and consuming semantically enriched content

6.10 Auto-generation and correction of content descriptions

6.11 Intra- and interlingual subtitling

For each of the functional epics, we have defined the following:

1. The preconditions which are required to be present before the functionality implemented in this epic can be executed, which in many cases consist of the output of other epics. This way, all epics are chained together to cover the entire functional process of media production and consumption as described in Section 4, which is illustrated in Figure 2. As such, the MeMAD prototype that is being built incorporates state-of-the art research results from work packages 2-5, it also supports a credible end-to-end media production process, and it illustrates how generated data and metadata can demonstrate its purpose across the entire implemented process.
2. The functional description which defines the features required to complete the epic's implementation. This description can be a single functional requirement, or it can consist of a list of features which each need to be built and combined to fully implement the epic's requirements. We also list further auxiliary requirements that apply to the required features but might not be strictly functional in nature.
3. We define how the successful implementation of the epic can be evaluated. In many cases, this consists of both an objective and subjective set of criteria, to ensure that tests can be repeated such that subsequent software versions can be compared in terms of performance, but also to ensure that (where applicable) the

[6] "User Stories Applied: For Agile Software Development", Mike Cohn, Addison Wesley Longman Publishing, 2004.

added value of the implementation is evaluated by a test panel in a way that could be difficult to test algorithmically.

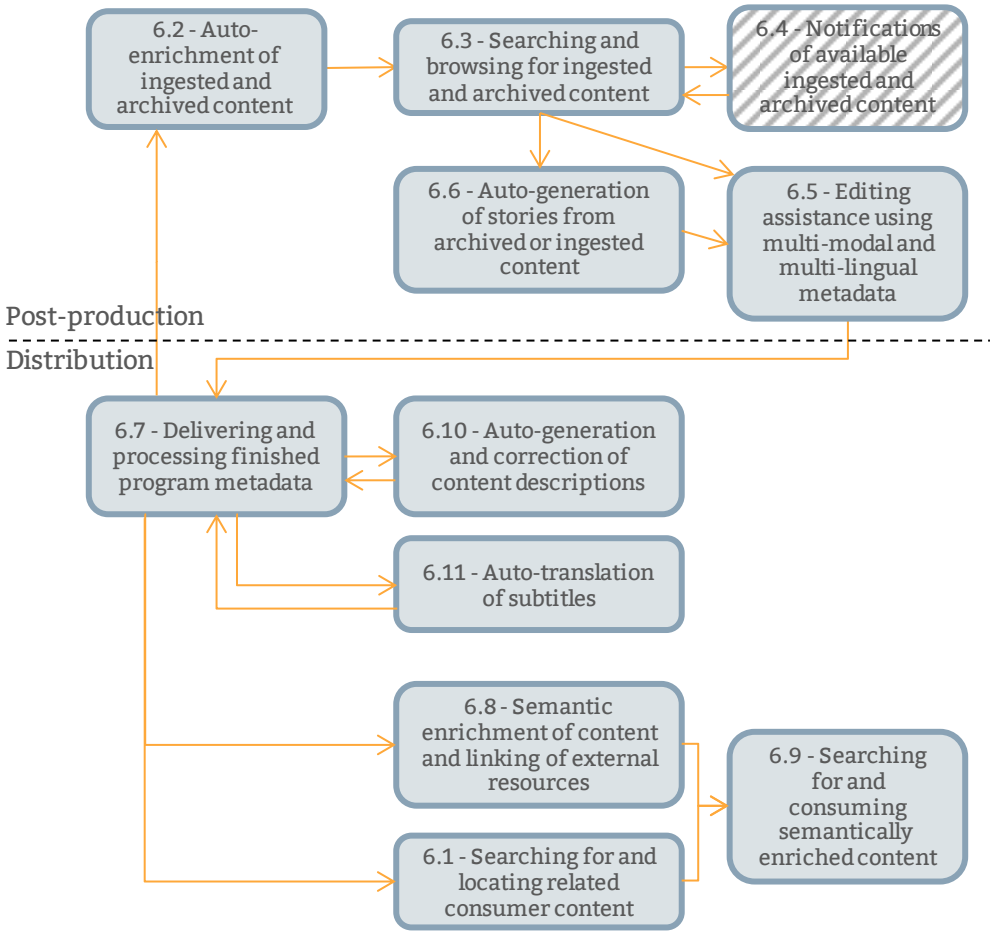

Figure 2: Coverage of the entire media production and consumption process by chaining functionality from the 10 implementation epics.

Proposed evaluation methodology

We propose a combination of techniques for the evaluation of the MeMAD prototype and its underlying components. Considering the complex end-user interactions and integrations with a variety of heterogeneous services, a single evaluation metric will not suffice to properly evaluate the success of the developed prototype.
With regards to functional requirements, we suggest using a combination of subjective end-user testing procedures and objective evaluation tests that can be automated.

For subjective user tests, which concern the human interaction with the prototype and its user interfaces, we will employ the following evaluation methods, building upon the evaluation plans constructed for the second round of end user testing (cf. D6.6):

1. Evaluation of user interfaces for executing specific tasks or workflows using the User Experience Questionnaire (UEQ)[7]. The UEQ has been designed and widely used to elicit users' impressions, feelings and attitudes towards interactive software products. It consists of 26 7-point Likert-type questions with a mid-point for neutral answers and variable labels, intended to measure both classic usability aspects and user experience aspects. Using this questionnaire will provide insights into the performance of a particular version of a user interface, but also demonstrate evolutions between subsequent software versions, which aligns well with the UCD principles followed in the prototype's development.
   Where necessary, the standard UEQ will be complemented by further sets of Likert-type questions, relating to usability and user experiences of the platform.
2. To ensure more subtle and intricate considerations about the developed user interfaces are captured too, we will implement think-aloud protocols with those test users that are suited to do so, i.e. having users verbalise their thoughts while they are carrying out a task[8]. Other users will be asked to participate in semi-structured *in-situ* interviews immediately after their practical participation in using the prototype platform.

For objective evaluations of the prototype functionality, objectively quantifiable tests will be defined that can be repeated as necessary (preferably using an automated testing suite such that no user interaction is required). We list a suggested set of these evaluation tests for each of the functional epics in the following subsections, steered by the findings reported in D6.6.
Note that we have deliberately opted to define evaluation metrics that measure the performance of the functionality implemented as a whole and from the perspective of the prototype system. Performance and accuracy tests of individual content or metadata analysis components are left to be evaluated in different work packages (2-5) using specifically optimized metrics.
More details concerning the execution of the final platform evaluation will be provided at a later stage in report D6.9.

[7] Cf. "Applying the User Experience Questionnaire (UEQ) in Different Evaluation Scenarios", Schrepp et al., In Proc. of the 3rd International Conference on Design, User Experience, and Usability (DUXU), June 22-27 2014. Also available at: https://www.ueq-online.org/.
[8] Cf. "Thinking aloud: Reconciling theory and practice", Boren, T. and Ramey, J., IEEE Transactions on Professional Communication, 43(3):261-278, October 2000.

6.1 Searching for and locating related consumer content

Covered user stories: 1.1.1, 1.1.2, 1.2.1, 1.2.2, 1.2.4

This epic groups those user stories that deal with searching and locating (parts of) content in a consumer content library, using search queries or through related content.

Preconditions:

- Program content enriched with a variety of metadata, with metadata available on the level of the entire program, and (if applicable) also on a more granular segment level. This can be delivered from Epic 6.7.

Functional description:

- Users can search directly for content (entire programs or program segments) thanks to metadata associated with all consumable content. Users can search across several dimensions and topics, incl.: persons, locations, time periods, subjects, etc.
- Users can access individual program segments if those have been described, in addition to accessing entire programs.
- Users can discover related content thanks to metadata enrichments added to consumed content. Properly typed relations can further refine the accuracy of these relations, for example, that two items are linked by the "location" relation, or by the "person" which they have in common.
- Users can search using combinations of parameters, i.e., to enable AND, OR and NOT operators on search queries. The search results returned should be ranked in terms of relevancy and should optionally also include related programs (or program segments). Additionally, users should be able to explicitly filter on specific relations (e.g., the "location" should be specified) through faceted search or query mechanism.
- Optionally, when consuming, users can skip those segments from a program that are not of interest to them, based on their preferences expressed as program metadata.
- Optionally, users should be able to find content by means of a visual search feature in which they indicate search queries by example, e.g., by indicating a face or landmark of their interest, which can then be localized in the content library. This feature can be supported by easily trained user-defined classifiers which are then used to look through the content library search index.

Evaluation:

1. The search user interface will be evaluated according to the methodology explained above. In particular, users should evaluate the ease of use through which they can enter search queries, by which they can browse search results, and how convenient it is for them to locate and start the consumption of content segments.
2. Objective evaluation metrics include:
    a. Given a search query (including various search operators and faceted filters), are all expected elements returned in the search results?
    b. Are the returned search results ranked according to the expected ranking?
    c. Can each of the relations between items be resolved as it should?

6.2 Auto-enrichment of ingested and archived content

Covered user stories: 1.2.3, 2.1.1, 2.1.2, 2.2.4.

This epic concerns the user stories that comprise the automated enrichment of content, at the early stages of media creation (upon ingestion into a production system) or when storing the media in an archive with the intention of later re-use. The epic also includes the functionality to manually correct auto-enriched content such that qualitatively adequate metadata is associated with content before it is committed to the media production process or to archive libraries.

Preconditions: None, except for video and audio content.

Functional description:

- Audiovisual content is tagged with multi-modal metadata to give insights on the kind of content that is represented.
- The enrichment consists of several processing steps, each contributed across the project by various consortium members:
    1. Perform audio classification and segmentation (cf. 7.1 and 7.2);
    2. Optionally, perform audio speaker identification on those audio sections that were classified as speech (cf. 7.4);
    3. Optionally, perform spoken language segmentation and classification to help steer subsequent speech recognition steps (cf. 0);
    4. Perform automated speech recognition (ASR), (cf. 7.3);
    5. Perform video captioning (cf. 7.7);
    6. Perform face detection and recognition (cf. 7.5);
    7. Execute named entity recognition on outputs from steps 2, 3 and 4. (cf. 7.9);
    8. Optionally, translate outputs from recognized entities to more common languages such as English or French to help in their disambiguation and linking to resources in common knowledge bases. (cf. 7.8);
    9. Disambiguate and semantically link recognized named entities (or their translations) (cf. 7.9);
    10. Finally, an auto-segmentation is performed on the bulk analytics metadata generated in the previous steps. Each resulting segment is associated with a summary of metadata grouped from the original content item metadata. The aim is to keep the most relevant or distinguishing metadata that clearly describes this segment (cf. 7.13).
- An intuitive GUI should be provided to correct the auto-generated content enrichment metadata (as produced by each of the components listed in the previous point).
- The enrichment process can incorporate complete program metadata at any stage in the enrichment chain to take advantage of prepared or externally delivered data. For example, subtitles that are created later can be injected into the enrichment process and be incorporated in the entity recognition and auto-segmentation phases. Such inputs can for instance be delivered by the processes implemented in Epic 6.7 (all relevant metadata), Epic 6.10 (content descriptions) and Epic 6.11 (subtitles).

Further requirements:

- Returned metadata should be timecoded and should support multi-lingual representation of the same metadata wherever applicable (e.g., named entities should be representable in different languages).
- Depending whether the enriched content is used in the context of an archive, or is part of the ingest process, the executed analysis processes could differ (e.g., because they are too time-consuming), or the manipulation and correction GUI could be implemented differently. The exact needs for this functionality will be determined through end-user interviews.
- Concerning the auto-segmentation, for each synthesized segment, summarizing terms can also be deduced based on the grouped metadata. E.g., a segment could be summarized as discussing "politics" or "climate", even if those terms are not explicitly mentioned. More details on how this functionality is supposed to work and which practical expectations we have from it are given in Section 7.13.
- Depending on the context of execution (e.g., ingest for short turn-over vs. archiving) long-running steps of the enrichment process could be skipped if timing constraints prohibit costly video or audio processing to be completed fully.

Evaluation:

1. The user interface for correcting auto-generated content enrichment metadata will be evaluated according to the methodology explained above. In particular, users should evaluate the ease of use through which they can gain an overview of the auto-generated enrichments and how conveniently they can add, update or remove enrichments. This aspect needs to be evaluated from a point of view of materials ingested during production, and for materials ingested as part of archival management.
2. The qualitative evaluation of the content enrichments created in this epic will be complicated due to the wide variety of underlying services integrated in this case. Despite this, performing this evaluation at the level of the prototype platform can provide an interesting bird's-eye view of the enrichment process which will be hard to obtain from evaluations of individual underlying components. As such, the main interest for evaluation lies with the results from multi-step processes such as the chain for audio processing with ASR and speaker recognition as the final steps, or video processing with captioning that incorporates facial recognition. Of particular interest will be the impact of error propagation on the end result: given errors in the first analysis step, how will this affect the accuracy of subsequent processes?
   Through the construction of a limited ground truth validation data set which contains curated (and manual) enrichments across all dimensions (transcripts, speaker recognition, face detection, disambiguation and finally segmentation) an objective measurement can be created to gauge precision and recall on the execution of pipelined analytics components. Important to learn in this case is which component weighs the most in delivering eventually accurate or unusable detection results.
3. In addition to a validation data set of which the enrichment can be evaluated by automated means, user test panels will also provide input on the accuracy they perceive when interacting with the enrichments presented by the MeMAD prototype platform. While subjective and less accurate, this evaluation technique will provide the consortium with more evaluation feedback than can be obtained from manually setup validation test sets. This complementary approach might also uncover feedback that would not be clear from strictly objective measurement methods.

4. Thanks to the availability of genuine 'legacy' archive metadata provided by INA and YLE as part of the media content data sets (cf. D1.2), a representative ground truth data set is available for evaluating the performance of automatically generated metadata vs. manually curated metadata. This allows us to measure to what extent this autogenerated metadata can replace the original metadata. Test panels will try to locate the same content using either legacy or autogenerated metadata in a set of A/B tests.
   Learning from the second evaluation round performed in late 2019 (cf. D6.6), we already observed that archive researchers tend to search content using queries attuned to the kinds of metadata typically entered by human curators, i.e., which use broader topics and generalizations than offered by literal autogenerated metadata (such as speech transcripts). This bias needs to be considered when performing the final set of evaluations.

6.3 Searching and browsing for ingested and archived content

Covered user stories: 2.1.3, 2.2.1, 2.2.2, 2.2.6.

In this epic, we group the functionality which concerns browsing and searching for content in professional media production or archive libraries. Unlike in Epic 6.1, searches can concern unfinished parts of media content with incomplete metadata and enrichment, which will require a different approach than for supporting consumer searches.

Preconditions:

- Enriched media content as delivered by Epic 6.2.
- Translated subtitles or transcripts as delivered by Epics 6.5 and 6.11.
- (Audio) content descriptions delivered by Epic 6.10.
- Semantic enrichments and links delivered by Epic 6.8.

Functional description:

- Users are provided with search functionality to search through an archive or media production system library of archived resp. ingested audiovisual content. Users can construct search queries using metadata that has been automatically added and possibly corrected. Users can search for detected topics, persons, speech fragments, music and image characteristics, detected emotions, etc. using named entities or free text queries.
- Users can also browse for content using a set of metadata facets such that they can drill down on lists of content beyond the filtering of a search query. These facets include descriptive metadata, but users can also use creation date or popularity as facets for browsing.
- Content will match search queries across all provided input metadata, including transcripts, subtitles, manual descriptions of content, visual content descriptions and audio descriptions, and even semantically related linked resources.
- Search results will include both entire programs or clips and those individual segments where the search query matches. For segments, a clear indication will be given how they relate to the entire program. Users will also be able to indicate the granularity of segmentation they wish to be presented with.

- Search results can be selected and stored for later use, or they can serve as input for other epics, incl. Epic 6.6.
- The exact preferences and search GUI specifications will be defined through future end-user interviews.

Evaluation:

1. The search user interface will be evaluated according to the methodology explained above. In particular, users should evaluate the ease of use through which they can enter search queries – even complex ones dictated by the complex needs of the production process, by which they can browse search results, and how convenient it is for them to select content segments for use in downstream production processes.
   As a learning from the second evaluation round (cf. D6.6), users will also judge how well they can distinguish between different granularities of metadata (i.e., low-level vs. high-level) and how this distinction can help in avoiding them getting lost in 'too much data' versus locating only those parts relevant to the search query. In particular, it will be important to observe how test users can work bottom-up from a speech transcript to entire segments of interest, or how auto-segmentation can deliver correct high-level concepts that effectively represent 'underlying' lower-level metadata (e.g., speech transcripts and face recognition results).
2. Objective evaluation can be performed similar to that defined for Epic 6.1, however with likely more complex search queries than those relevant for consumer scenarios.

~~6.4 Notifications of available ingested and archived content~~

Covered user stories: 2.1.4, 2.2.3.

This epic covers functionality that brings content to the users as opposed to delivering results as a direct response to user requests. By setting up notifications when suitable content is available, users are kept aware of the presence of this content without actively needing to search for it.

As both relevant user stories for this epic have been discarded from implementation in the project, this epic is also discarded.

6.5 Editing assistance using multi-modal and multi-lingual metadata

Covered user stories: 2.1.5, 2.1.6

This epic combines those user stories that focus specifically on functionality to aid audiovisual content editors with their daily tasks, including triaging content in preparation for editing, and understanding foreign language content during the editing process.

Preconditions:

- Enriched media content as delivered by Epic 6.2.
- Selections of media content made in Epic 6.3.

- Optionally, this epic can also incorporate complete program metadata from an archive context, as delivered through Epic 6.7.

Functional description:

- Content metadata that is deemed relevant for editors is made available to editors in a convenient way, e.g., by exporting it per (batch of) items to their editing environment such that it can be searched for in that working environment.
- Relevant content that has been searched for and selected for the edit (cf. Epic 6.3) can easily be imported into the craft editing environment, along with the enrichment metadata.
- The associated metadata is shown in the editing environment in the most optimal way, e.g.:
    - In a table of content items for content that is applicable to an entire video or audio clip;
    - Along the clip's timeline for time-varying metadata such as transcripts of spoken dialogue.
- In order to support user story 2.1.6, transcripts associated with the selected media are processed by a translation component to the editor's preferred language, and then made available as an additional layer of metadata (in addition to the original dialogue transcript data).
- Automatically translated transcripts can be corrected by means of a dedicated user interface such that translations which show an unworkable amount of errors or inconsistencies can be corrected before they are sent off to a video editor.

Further requirements:

- When translating transcript data, timing information provided as input to the translation process should be retained wherever possible in the translation output such that MeMAD user interfaces can still allow users to navigate the audiovisual content by means of the translated text. This will also allow GUIs to show e.g., side-by-side various translations for comparison or correction purposes.

Evaluation:

1. Given that the majority of the end-user's work will be carried out in a craft editing environment outside of the project's control, the presented user interface can only be evaluated to a certain extent. However, users can evaluate how well the presentation of content metadata makes use of the editing and material triaging features provided by the editing environment.
   To a lesser extent, the user interface that is provided for corrections of (translations of) transcripts can be evaluated as described above, with a focus on how multiple translations are presented and how convenient the correction process is.
2. Objective evaluation metrics include a subset of those defined for Epic 6.11, focused on the aspect of translation of ASR transcripts.
3. Subjective evaluation of the quality of provided metadata will be complementary to the evaluation in Epic 6.2 as the users evaluating this functionality will have a different role (as editor), and hence different priorities and needs, as in the other case.

6.6 Auto-generation of stories from archived or ingested content

Covered user stories: 2.2.7

Researchers and editors can create a search query that results in an automatic story constructed from relevant and related archive content segments. They receive a story proposal where they can efficiently change the selection and the order of the footage. Given the wide range of possible interpretations that can be given to what exactly the functionality of this epic could be, we provide some initial steering based on the current state-of-the-art in research on this topic and expectations of members from the media production industry.

Interest in the topic of auto-generation of rough stories was confirmed by multiple ECG members. Additionally, it was deemed an interesting trajectory to take for further building upon the segmentation work aimed at improving content retrieval on one hand and TV moments detection work planned for WP3 for which a clear use case on the content creation side was not yet defined. A way of providing a practical application to the Media Memorability challenges being worked on in the research community would be that if the memorability algorithms can roughly predict which sections of media will be memorable to an audience, those segments should be suggested to content creators as a starting point for constructing equally memorable stories. Taking clues from the capabilities of the state-of-the-art on detecting memorable media segments (as also discussed in D3.2), the intent will not be to assemble fully completed and polished stories ready to be shown to audiences, but rather provide sets of media segment suggestions to content makers. These suggested segments are then meant to be tweaked and edited further manually in a video editing environment before they can be distributed to consumers. In the case that the auto-selection algorithms do exceed expectations (for example, by replicating an original program's segment ordering) we will actually allow the exposure of auto-generated stories to test audiences to gauge end user satisfaction and future areas of research.
Further details and fine-tuning of the designs for automated story building will be done in collaboration with members of the ECG.

Preconditions:

- Metadata-enriched content as delivered from Epics 6.2, 6.11 and 6.10.
- Optionally, selections of media content made in Epic 6.3.

Functional description:

- Based on metadata-enriched content, story suggestions can be presented to both end users in media production and consumption.
- Suggested stories are assembled based on a set of search queries, or selection results from Epic 6.3 and are optionally constructed according to a limited set of templates (e.g., documentary item, best-of-moments, etc.). This process first uses the auto-segmentation performed in Epic 6.2 (also, cf. 7.13) to gather further insights into the media's content and where to potentially pick relevant items from. It then uses relevant moment detection (cf. 7.14) to prioritize the likely most interesting segments, the result of which is then presented to the end user.
- Suggested stories can be imported into a craft editing environment such that they can be modified and given a proper narrative structure before publication.

Further requirements:

- The goal of this functionality is not to generate finished montages of content. Rather, the aim is to present the editing process with a story template composed of potentially relevant source materials from which editors can cut together a finished program item.

Evaluation:

1. As this epic will reuse user interfaces from Epic 6.3 no additional evaluation is foreseen at the moment regarding the usability aspect of the story generation itself (which will feature only limited user interaction within the MeMAD platform), the generated stories themselves, however, can be evaluated according to the next point.
2. To the extent that objective evaluation metrics can be defined in this case, the validity of the proposed story can be measured by the ratio of relevant vs. non-relevant chosen clips or segments in an auto-generated story (when generated from a dataset for which a validated ground truth has been manually curated). In those test cases not described by a validation data set, the decision whether content "is relevant" within a story will be left to the expert user. Similarly, the ordering and duration of chosen segments can be given a subjective score by these expert users. In all, further subjective user questions can be used to gauge the overall usefulness of generated stories.

6.7 Delivering and processing finished program metadata

Covered user stories: 2.3.1, 2.3.2, 2.3.3

Preconditions:

- A finished program, assembled from pieces of enriched media content and defined by an Edit Decision List (EDL).
- Content metadata associated with the used pieces of media content, as produced in from Epics 6.2, 6.11 and 6.10.

Functional description:

- As a first step, in order to grasp the structure of the program, the EDL is processed to learn how it is constructed from pieces of source content.
- From the used pieces of source content, the related metadata is then retrieved and combined into a single data set that can be exported from the prototype platform and be used as input for another similar system, or for data aggregation about the delivered program.
- The reverse operation of the export function is also implemented to demonstrate the ability to process a combined set of metadata descriptions for a program and to perform analytics on this data set, for example, the following:
    - Metadata is can be stored and queried using queries and combinations not envisioned at production time;
    - Metadata from different programs can be related to each other;
    - Search queries can be made to retrieve information across programs.
    - Metadata from a single program can be combined from different sources and can form a comprehensive data set that can be queried in its entirety.

Further requirements:

- To maintain the proper focus in this project, this Epic will begin using an existing EDL which describes a program's structure in detail. In many real-world scenarios this EDL would not be available and content detection techniques could be used to reconstruct the structure of the original program. This is outside the scope of the project.

Evaluation:

1. As there is no explicit end user interface that is presented to manage or visualize this process no explicit evaluation is foreseen for this aspect.
2. Objective evaluation metrics include:
    a. In the case of EDL processing, successful processing can be measured by tracking the number of correctly retrieved content elements from the EDL source file.
    b. Validations to verify that all relevant metadata present in the source system is included, then correctly exported as first step, and then processed and available for further processing in the target system after an import-and-export operation in a second phase.
    c. Where imported metadata on one hand and existing metadata on the other hand overlap in the target system, it can be measured to what extent each metadata item is correctly linked together after the import operation.

6.8 Semantic enrichment of content and linking of external resources

Covered user stories: 3.2.1, 3.2.2, ~~3.3.1~~.

Apart from enrichments that take place to describe the audiovisual content as accurately as possible, this epic deals with another step of enrichments, based on the semantics deduced from the content (and its other enrichments). The aim of this second set of enrichments is to provide users with a better context surrounding the content in the hope of improving user consumption experiences.

Preconditions:

- Content that has been enriched by content analysis tools and that has been semantically disambiguated and linked with relevant resources, as produced by Epics 6.2, 6.11 and 6.10.

Functional description:

- Using media content that has been enriched in previous processes, content items are further extended with associations to external resources. Such linking includes (the exact set of enrichments and linking will be determined in T3.3):
    - Making references to news articles automatically retrieved and promoted by the broadcaster;
    - Including visualizations or raw data of data analytics performed on social media concerning the topics of the content;
    - Linking to scientific knowledge bases about the subjects of the content;
    - Linking to other related resources, e.g., by employing publicly available knowledge graphs (e.g., Wikidata or DBpedia).
- Linking is done on three levels:
    - Some semantic relations will be placed on the program level and are applicable to the entire program;

    - Some relations will be placed on (temporal) segments of the program;
    - Some relations will be placed on spatial elements within the program (e.g., linking an animal visible in the image with resources about that animal).
- Resource linking is also setup in reverse such that the linked resources can be used to find related items. For instance, if two media segments are linked with the same news item, they can both be found through that news item.
- As part of the semantic enrichment of media programs, optionally, the media content – or at least those metadata that give an insight into its content such as ASR transcripts or resources that have been linked to it – are fed through a service that can analyze this content for truthfulness analysis. The truthfulness score returned by this service is incorporated as an external resource which can then be presented to consumers.

Evaluation:

1. As there is no explicit user interface that is presented to manage or visualize this process no explicit evaluation is foreseen. The results of the enrichment process will be evaluated as part of the Epics 6.1 and 6.9 evaluations.
2. To the extent that objective evaluation metrics can be defined in this case, the validity of the semantic enrichment can be measured by the ratio of relevant vs. non-relevant linked items proposed (when generated from a dataset for which a validated ground truth has been manually curated). We propose the adoption of a relevancy score to allow a nuanced evaluation of the algorithm's output.
   In those test cases not described by a validation data set, the score of whether enrichments and linked media "are relevant" to be associated with a content item will be answered by the consumer user judgement.

6.9 Searching for and consuming semantically enriched content

Covered user stories: 3.1.1

Once content has been semantically enriched and linked to external resources, it also needs to be located and consumed by end users, which is the subject of this epic.

Preconditions:

- Content that has been semantically enriched and linked with relevant resources (cf. Epic 6.8).

Functional description:

- Users can browse content starting from a content item or program and they are then shown the linked resources of, or inversely, they can browse linked resources (e.g., a list of news articles published by the broadcaster) and be shown the related content items.
- When watching a media program, users are interactively prompted about relevant related content when that content applies only to a part of the program (e.g., a segment or spatial part, cf. the previous epic).

Evaluation:

1. The search user interface will be evaluated according to the methodology explained above. In particular, users should evaluate the ease of use through which they can find the desired media content by using the provided semantic enrichment and linked resources. Additionally, they should rate the enhancement

in media consumption experience with linked resources as opposed to consuming the same content plainly. Finally, users can indicate to what extent the linked resources and semantic enhancements themselves provide accurate background information about the content (i.e., whether relevant resources were linked, whether only those resources were linked that represent a correct interpretation of the source content and finally whether the provided links actually proved to inform users of worthwhile additional information).

6.10 Auto-generation and correction of content descriptions

Covered user stories: 4.2.1, 4.2.2, ~~4.1.2~~.

This epic combines the implementation of automated content (audio) description generation.
Based on observations from WP5 in D5.1 and D5.2, the primary focus of the content descriptions generated in the MeMAD project will be to augment media for easier content retrieval, e.g., from archives. Secondly, but more experimentally, the content descriptions will also be trialed in an audio description context, however with limited expectations given the state-of-the-art performance of using video captioning algorithms for the generation of more advanced narratives. As this is especially the case for dramatic and fictional content, we will also experiment with the generation of (audio) content descriptions for other program formats, such as current affairs or lifestyle programs which exhibit more straightforward narrative structures.

Preconditions:

- Enriched media content as delivered by Epic 6.2 and 6.11.
- Optionally, other delivered program metadata such as program scripts can be used as produced by Epic 6.7.

Functional description:

- Using audio transcripts, speaker and gender identification, and visual content metadata incl. video captions, and recognized persons, content descriptions are generated that narratively describe the visual characteristics of the video content.
- The generated content descriptions should incorporate the findings of WP5 (cf. work done in T5.2 and T5.3) such that they resemble more human-like content descriptions.
- The generated content descriptions should be time-aligned such that it can be identified to which temporal part of the content each description applies.
- Generated content descriptions can combine multiple elements from the input metadata (captions, transcripts, etc.) and “re-write” them in such a way that a better or more informative narrative is obtained which gives readers a better insight into the semantics of the (audio)visual content.
- Content description generation will incorporate multimodal translation such that descriptions in a variety of languages can be produced simultaneously.
- Optionally, content description generation can take into account additional program metadata delivered along with the audiovisual content, such as production scripts which contain prepared knowledge about the program and which could give insights into the semantics of the program that are not

obviously deducted from the spoken narrative (transcripts) or video images (video captions).

- As optional extension, the generation of content descriptions can be optimized in such a way that they fit in between, and provide complementary information to, existing dialogue and relevant sound effects. This way, they could be used as a rudimentary form of audio content descriptions if rendered to an audio signal using a text-to-speech (TTS) system.
  Due to the complexity of this goal, as also illustrated by the findings described in D5.1 regarding the subtleties and intricacies of audio descriptions, we define it as an optional goal which will likely be executed with limited scope regarding the content formats for which audio descriptions can be delivered. For example, we will begin with short-form content (e.g., as delivered to many social networks or to news websites) and work our way up towards more complex narratives (e.g., documentaries and current affairs programs). The implementation of this goal also greatly depends on the performance that can be achieved from the video captioning and multi-modal translation components.
- A specialized GUI must be provided to review and correct auto-generated content descriptions such that they can be delivered with sufficient quality to actually help test panels to better understand consumed content. The exact requirements for this GUI will be determined through interviews with professional producers (from the content archive domain, from the content publishing domain, and from the audio description domain).

Further requirements:

- With respect to D6.4, we removed the further requirements regarding live audio description workflows as these user stories, and in particular, user story 4.1.2 are left out of the implementation.

Evaluation:

1. The user interface for correcting auto-generated content descriptions will be evaluated according to the methodology explained above. In particular, users should evaluate the ease of use through which they can correct existing auto-generated content descriptions and how conveniently they can add completely new elements in case the automated algorithms produced nonsensical descriptions.
2. Concerning the evaluation of the generated content descriptions themselves, we suggest the following rough set of metrics, which are to be discussed as part of further work in T5.3 and T5.4:
   As there are no standard evaluation methods for audiovisual content description or audio description, we will use a range of sources, including ISO/IEC TS 20071-21:2015 (Guidance on audio description), YLE's guidelines on audiovisual content description, national audio description guidelines from different countries and our observations from the comparative analysis of human and machine-generated content descriptions in the earlier project phases, in order to develop a) a small number of models of audiovisual content description (for different purposes) and b) two sets of evaluation criteria in relation to each model: the first set will be a simple set of criteria focused on the grammaticality, semantic accuracy, completeness (to the required level, according to the model) and relevance of the descriptions. The second set will be a more elaborate, fine-grained version. The criteria will be used to evaluate a test set of audiovisual content descriptions produced under different conditions (as per user stories, i.e.

automated/unsupervised, automated/supervised/post-edited, human descriptions). The simple data set will be given to human evaluators who will be asked to rate the different types of descriptions. The second, more elaborate set of criteria will be used in a more comprehensive qualitative evaluation that will elicit and examine characteristic strengths and weaknesses in the descriptions and will lead to recommendations for different purposes (e.g. when is unsupervised creation of descriptions possible, when is supervised/post-edited descriptions possible etc.).

6.11 Intra- and interlingual subtitling

Covered user stories: 4.3.1. 4.3.2, 4.3.3, 4.3.4, 4.1.4, ~~4.1.1, 4.1.3~~.

This final epic concerns the implementation of various aspects of (automated) subtitling, including the support for same-language (intra-lingual) subtitling and auto-translation of subtitling (interlingual subtitling), with its supporting user interfaces.

Preconditions:

- Optionally, existing subtitles are available in a limited number (possibly only one) of languages, e.g., delivered through Epic 6.7.
- Optionally, (corrected) ASR transcripts in the case that no subtitles are available, from Epic 6.2.

Functional description:

- If no existing subtitles are available, audio transcript data is used as the source for translation to a set of languages.
- In that case, subtitles are generated from the translated transcript by applying a set of 'spotting' and formatting rules to generate subtitles that are optimized in terms of placement, maximum textual length, duration, etc.
- In the case that source subtitles are available, the translation can be done either using the existing subtitle timing as a template for timing, or the subtitle text can be translated and be re-divided into actual subtitles by re-applying the spotting and formatting rules. The exact needs for this functionality will be determined through end-user interviews.
- Generated subtitles can be visualized and corrected using an optimized GUI such that defects in generated subtitles can be corrected and professional-quality subtitles can be delivered by the MeMAD system. The exact requirements for this GUI will be determined through interviews with professional subtitlers and translators.
- As a baseline feature set for this functional case, the generation of subtitles can also be performed in an intra-lingual setting by generating subtitles from the original audio transcript. This serves to implement user story 4.1.4.

Further requirements:

- The set of languages to translate (transcripts or subtitles) from is:
    - Finnish
    - Swedish
    - English
    - French
    - Dutch

    - Norwegian (Optionally)
- The set of languages to translate to is:
    - Swedish
    - Finnish
    - English
    - French
    - Dutch
- As is the case with Epic 6.5, when translating transcript data, timing information provided as input to the translation process should be retained wherever possible in the translation output such that this timing information can be used for the generation of properly timed subtitles.
- To aid the translation, visual video captions, manual annotations, or literal speech transcripts (each delivered by Epic 6.2) can be provided as context to the translation process, whenever available.
- With regard to D6.4, we removed the requirements to support live subtitling workflows, as these user stories (i.e., 4.1.1 and 4.1.3) have been discarded from implementation.

Evaluation:

1. The user interface for authoring and correcting auto-translated (and auto-generated) subtitles will be evaluated according to the methodology explained above. In particular, users should evaluate the ease of use through which they can correct auto-translated subtitles, how easily then can adapt subtitle timings and split and merge subtitles in case of misalignments, and how conveniently they can add completely new self-translated subtitles in cases when automated translations fail.
2. Regarding the evaluation of generated translated subtitles, the following elements should be incorporated:
    a. An objective metric to measure the ratio of correctly vs. incorrectly translated words in a set of subtitles to gauge the overall translation performance of the translation components. This evaluation procedure can be setup using a validation dataset for which the ground truth of acceptable translations is well-known.
    b. A set of metrics to measure the appreciation of the translated subtitles, scored subjectively by testing panels of both professional subtitlers, regular content consumers and hearing-impaired consumers. These metrics should gauge the correctness of the translation, but also indicate the eloquence of the generated subtitles. While individual scores might be difficult to judge individually, they could be compared with scores given to professionally made subtitles or to different generations of auto-translated subtitles of the same content (e.g., constructed by subsequent versions of underlying translation and ASR algorithms).
    c. As with the evaluation of Epics 6.2, 6.5 and 6.10 the impact of error propagation between automated enrichment services should be measured. In this particular case, it should be measured what the impact of ASR failures is on the subsequently translated subtitles.
    d. Extending the results of the second round of evaluations (cf. D6.6), process efficiency improvements must be measured in the case of subtitling. It is the most clear-cut use case and the valid process execution improvements can easily be defined and measured, incl. time required to finish broadcast-quality subtitles starting from e.g., existing (other-language)

subtitles, or ASR-generated audio transcripts. Related to point (a), another metric is the number of post-editing actions required to deliver a finished result. Finally, based on the results from point (b), efficiency gains can also be measured by the ratio of auto-generated subtitles that can be delivered to audiences with minimal to no post-editing, versus the total amount of media content in need for subtitling.

7 Content and metadata processing component requirements

Conceptually, the broad picture of the intended integrated MeMAD prototype platform is shown in Figure 3. The platform provides a unified view on audiovisual content and provides GUIs for executing production tasks as well as coordinating various media and metadata processing tasks. Workflows are executed by the platform, and the tasks that comprise those workflows are then executed by processing components delivered by each of the work packages 2-5. To support these processing components, the platform offers storage of the source audio and video content, along with options to transcode to other audiovisual formats to help in easier processing, and it will also store audiovisual content and various forms of metadata.

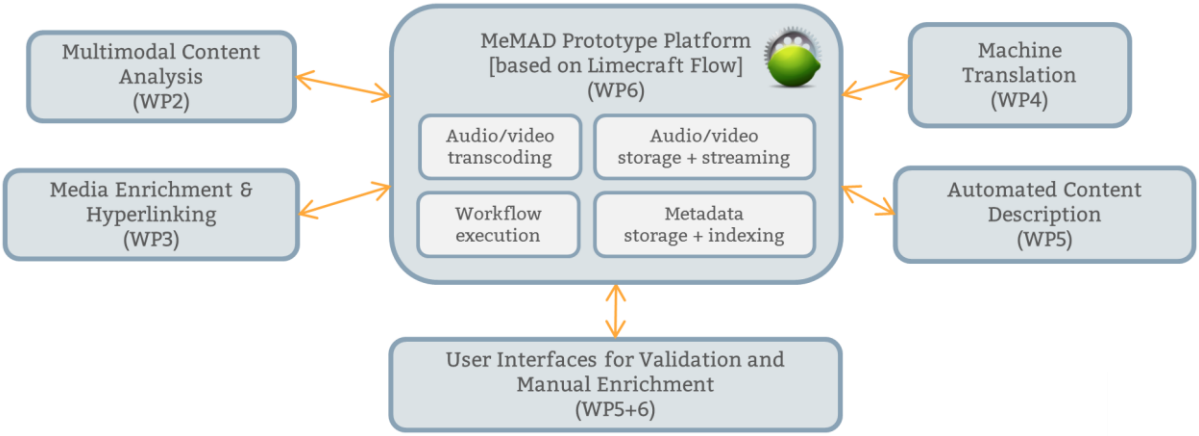

Figure 3: Conceptual overview of the MeMAD integrated platform.

We note that the core of the prototype platform will be based on the Limecraft Flow product, which is already commercialized in a Software-as-a-Service model by Limecraft. For most of its features, the platform requires no local installation of software applications and can be used from a standard web browser.

The user stories and implementation epics introduced in the previous sections have been defined from the point of view of the prototype platform. The platform implements core functionality such as user interfaces, content library functions, search capabilities, storage mechanisms for all forms of metadata required by the media production process. In addition, to successfully implement all epics described, the platform relies on external functionality provided by components implemented in work packages 2-5. To clarify what exactly are the features required from each component, a break-down is now provided per identified component, derived from the implementation requirements defined in the previous section. This breakdown lists the inputs and outputs of the component, a set of potentially relevant data context to improve the performance of each component, and finally lists the functionality required of each processing tool.

7.1 Audio segmentation

Audio segmentation segments an audio signal into audio sections of a limited set of audio classes (speech, music and silence) to allow for processing the audio signal more efficiently by downstream processing services. This service will be developed in T2.1.

Inputs:

- Audio signal (IO2).

Outputs:

- Audio segmentation (IO3).

With additional context:

- None.

Requirements:

- This service outputs timed segments of audio, each of which identified by one of the following audio classifications:
  - Speech (male or female);
  - Music;
  - Silence.
- Optionally, a confidence score is provided with each detected segment.
- This service is language-agnostic and can operate without being notified of the language spoken (if any) in the audio signal.

7.2 Audio classification

Audio classification classifies an audio stream into segments of a wide set of audio classes (musical instruments, noises, animal sounds, etc.). This service will be developed in T2.1.

Inputs:

- Audio signal (IO2).
- Optionally, an audio segmentation (IO3).

Outputs:

- Audio classification (IO4).

With additional context:

- None.

Requirements:

- This service outputs timed segments of audio, each of which identified by a wide variety of audio classifications, e.g., those defined in the Google AudioSet[9]. One segment can be assigned multiple classes of audio.

[9] "Audio Set: An ontology and human-labeled dataset for audio events", Gemmeke et al., Proc. of IEEE ICASSP 2017, and available at: https://research.google.com/audioset/ontology/.

- Optionally, a confidence score is provided with each classified segment (and for each class assigned to that segment, if applicable).
- This service is language-agnostic and can operate without being notified of the language spoken (if any) in the audio signal.

7.3 Automated speech recognition (ASR)

Automated speech recognition of audio signals (incl. speaker diarization), developed in T2.1.

Inputs:

- Audio signal (IO2);
- Speech segmentation information, optionally including language segmentation and classification information (IO3).

Outputs:

- Timed speech transcript (IO5).

With additional context:

- Optionally, prepared transcripts (e.g., from manual transcription and without associated timing information).
- Optionally, a custom dictionary of terms is provided to help the ASR component to detect domain-specific words and terms (e.g., person names or other relevant named entities) that are not present in general-purpose language models.

Requirements:

- The ASR service is given a language parameter as input, which specifies the expected language of the speech present in the audio signal. Alternatively, and optionally, the ASR service can also be provided with an extended version of the audio segmentation (IO3) in which the individual speech segments have been assigned a language field. This field can then be used by the ASR service to selectively transcribe only parts of the audio signal supported by the given ASR service.
- Speech recognition should be provided in the following languages to ensure that the available data sets can be sufficiently processed:
    - Finnish
    - Swedish
    - English
    - French
    - Dutch
    - Norwegian (optionally)
- The transcript is output as fragments of text as spoken, with per-word timing information and wherever possible also confidence scores for each recognized word. The fragments are output in such a way that speaker turns are grouped into single fragments. Additionally, each speech fragment is associated to a speaker. Fragments from the same speaker (as determined by the ASR service) are assigned the same speaker label.
- Optionally, alternative transcriptions can be provided (with their respective confidence scores).

- When prepared transcripts are provided as additional context, the provided transcript are to be used as ground truth of the spoken text and this transcript is then re-used but time-aligned with the audio signal.

7.4 Speaker recognition

Speaker recognition identifies patterns in speech audio signals and assigns them to a speaker that can be associated with that pattern. It produces an identification of speakers from the given audio signal.

Inputs:

- Audio signal (IO2).
- Audio segmentation (IO3).
- Optionally, a timed speech transcript (IO5).

Outputs:

- Speaker identification (IO6).

With additional context:

- This component is supported by a library of speaker profiles from which the recognition can be performed.

Requirements:

- Using an audio signal and a segmentation that identifies the speech segments in the audio signal, this service outputs identifications of speakers for each segment a speaker was correctly recognized. The speaker identification is output utilizing the same segments as provided as input, but extended with an identification of the speaker in question. A confidence score is provided along with the identification.
- Optionally, if multiple candidate speakers have been identified, alternatives are also output (again, with a confidence score to gauge the prediction differences for each candidate match).
- The optionally provided transcript can be utilized as a hint about the groupings of speech into speaker turns and identification of common speakers in the audio signal without specifying who the speaker actually is, which can then be provided by the speaker identification service.
- This service is language-agnostic and can operate without being notified of the language spoken (if any) in the audio signal.
- The service should easily accept additional speaker profiles such that they can be quickly added to the service's database without relearning a significant part of its underlying identification models.
- Optionally, a parameter can be provided to the service which lists a set of potential speakers in the audio signal, which can help reduce the search space considered by the identification service.

7.5 Face detection and recognition

As with speaker recognition, face recognition will identify face patterns in video signals and assign them to a known person's facial features. It produces an identification of a person from the given video signal. This component will be delivered in T2.1.

Inputs:

- Video signal (IO1).

Outputs:

- Visual person identification (IO7).

With additional context:

- This component is supported by a library of person facial profiles from which the recognition can be performed.

Requirements:

- Using a video signal, this service outputs identifications of persons for each relevant segment of video where the person's face was detected and identified.
- For each segment of a detected person, at least a fixed average spatial location within the video image is provided as part of the output. A full trajectory of (spatial) locations along the video's temporal axis can be provided as an extension. Ideally, spatial locations take the form of a bounding box such that both the position and size of the detected face are described. Considering that segments describe a single person, segments that overlap in time can be part of the output.
- A confidence score is provided along with the identification. In case no person could be identified for a given detected face, the detected face is still output as a segment, but without an associated person identification.
- Optionally, if multiple candidate persons have been identified, alternatives are also output (again, with a confidence score to gauge the prediction differences for each candidate match).
- The service should easily accept additional facial profiles such that they can be quickly added to the service's database without relearning a significant part of its underlying identification models.
- Optionally, a parameter can be provided to the service which lists a set of potential persons in the video signal, which can help reduce the search space considered by the identification service.

7.6 Shot-cut detection

Shot-cut detection involves the detection of transitions between logical 'shots' in a video signal. These shots represent different camera viewpoints or different subjects recorded in the image. Shots-cuts are typically the result of editing decisions in the program's making process, but unfortunately, the position of these cuts is typically not stored beyond the post-production process, leading to the need of re-detection of these features afterwards. This component is part of the prototype platform, delivered in T6.2.

Inputs:

- Video signal (IO1).

Outputs:
- Shot-cut boundaries (IO8).

With additional context:
- None.

Requirements:
- Using a video signal, this service outputs the position in time when a shot-cut likely occurs, based on abrupt changes in image characteristics.
- A confidence score is optionally provided along with each detected shot-cut.

7.7 Video captioning

The video captioning component produces human-like descriptions for video content, taking into account both the visual and aural domains and referring to the recurrent objects and persons in human-like intelligent ways. In addition, the video captioning is built to inherently incorporate the temporal dimension of video by producing video captions that are more relevant than subsequent still image captions. This service is developed as part of T2.3.

Inputs:
- Video signal (IO1).
- Shot-cut boundaries (IO8).
- Visual person identification (IO7).

Outputs:
- Natural language video captions (IO14).

With additional context:
- Transcripts (IO5).
- Subtitles (if available from manual sources, IO12).
- Manually added tags by users (IO10).

Requirements:
- Captions are returned with timing information to identify start and end of when the caption applies.
- Captions are returned with spatial information to identify where the caption applies (if applicable).
- If recognised faces are provided, the captions should incorporate this information in the captions.
- Objects described should be output in a disambiguated way in the captions (so that there's no guessing about which the exact entity is meant).

7.8 Machine translation

A machine translation component will be front-and-center within the MeMAD prototype, serving a variety of automated translation tasks, translating many types of metadata between languages with the aim of overcoming language barriers in the project's media production and consumption use cases. This translation service is built in T4.3.

Inputs:

- Text fragments from various sources, including subtitles (IO12), speech transcripts (IO5), video captions (IO14) and manually-made comments and named entities (IO10).

Outputs:

- Translated Text fragments which can be re-used for a variety of destinations, similar to the types of inputs (see above). (IO9).

With additional context:

- Optionally, additional text fragments are provided besides the source input to provide more context to the translation algorithm and potentially improve the translation accuracy. These text fragments can be derived from a variety of metadata, incl. subtitles (IO12), speech transcripts (IO5), video captions (IO14) and user-made comments and named entities (both as IO10).
- Optionally, the purpose or text domain of the translation can be provided such that an optimized translation model can be employed.

Requirements:

- Translation should be provided to and from the following languages as part of the pre-visioned project prototypes and given the provided testing data sets:
  - Finnish
  - Swedish
  - English
  - French
  - Dutch
  - Norwegian (Optionally)
- The translation service is given source and output language parameters as input.
- As part of the translated output, the service provides a mapping between the source text and the output text, which allows other services in the system to trace back the evolution of pieces of text from source to translation. Note that this is a best-effort requirement; the service might be implemented in such a way that only non-descending positions are given in the output.
- When translating smaller text fragments, the service can be provided with additional context to help improve the translation.

7.9 Named entity recognition and disambiguation

Named entity recognition processes a text input, detects words that represent relevant named entities (e.g., persons, concepts, geographical locations, etc.) and disambiguates these found entities with markup to identify the exact identification of the entity (amongst a number of candidate entities that share the same name, or in case only a part of a name is provided in the source text) and type of entity recognized according to a fine-grained taxonomy. This component is delivered by T3.1.

Inputs:

- Text fragments from various sources, including subtitles (IO12), speech transcripts (IO5), video captions (IO14) or user-made comments (stored in the prototype platform and exchanged as free-text timed comments).

Outputs:

- Text with detected and disambiguated named entities (IO10).

With additional context:

- Optionally, additional text fragments are provided besides the source input to provide more context to the entity recognition and disambiguation algorithm to potentially improve the detection accuracy. These text fragments can be also derived from a variety of metadata, incl. subtitles (IO12), speech transcripts (IO5), video captions (IO14) and user-made comments and named entities (both as IO10).
- Optionally, hinted items can be provided to help the detection and disambiguation of entities. These hinted items could be obtained from manually added annotations.

Requirements:

- Entity spotting should be provided for the following languages as part of the pre-visioned project prototypes and given the provided testing data sets:
    - Finnish
    - Swedish
    - English
    - French
    - Dutch
    - Norwegian (Optionally)
- As seen from the user stories, the following are entities types required to be described, if they can be detected:
    - persons;
    - objects/nouns;
    - time periods;
    - places;
    - affiliations and political orientations;
    - actions;
    - overall topics or subjects (e.g., “economics”, “politics”)
    - environmental characteristics.
- The service outputs detected entities as mentions with regard to the original text, identifying the start and end character positions where the entity was found.
- The service is given the source text’s language as an input parameter.
- A confidence score is optionally associated with the disambiguation of an entity to express the probability that the text actually represents this disambiguated entity. Multiple disambiguation candidates can be returned in the output (if applicable), but in this case, their respective confidence scores need to be provided such that an evaluation of the candidates can be made outside the service.
- Disambiguated entities are output by means of a unique identifier which can be used to lookup further information concerning the entity through external

sources such as DBpedia[10] or Wikidata[11]. In addition, an optional disambiguated label can be provided for visualization purposes (to avoid mandatory access to external systems when visualizing the output in question).

7.10 Semantic enrichment

Semantic enrichment will extend disambiguated text fragments with links to additional resources, including news articles automatically retrieved and promoted by the broadcaster, related videos and video excerpts and visualization of data analytics performed on social media enhanced by structured data extracted from knowledge graphs such as Wikidata and DBpedia. This component will be delivered in T3.3.

Inputs:

- Text with detected and disambiguated named entities (IO10).

Outputs:

- Semantically enriched text with detected and disambiguated named entities and links to related resources (IO11).

With additional context:

- None.

Requirements:

- The input is enriched with a variety of related resources, each of which is identified through a URI, and ideally also a link type (e.g., “article”, “game”, “discussion forum”, etc.). Related resources are also linked to the source text by identifying the part of the text where the link is applicable (in the case that the linking is not done through named entities already identified in the text). Linked resources that apply to the text item in its entirety should be flagged as such.

7.11 Subtitle generation

Automated subtitle generation creates subtitles from timed transcripts or similar timed text input (which has not yet been split and formatted into correct subtitles), taking into account a set of ‘spotting’ rules that dictate the permitted duration, length and styling parameters for generated subtitles. An updated version of this component is developed in task T6.2 in collaboration with T4.3.

Inputs:

- Timed speech transcripts (IO5)
- Shot-cut boundaries (IO8)

Outputs:

- Subtitles (IO12).

[10] Cf. https://wiki.dbpedia.org/
[11] Cf. https://www.wikidata.org/wiki/Wikidata:Main_Page

With additional context:

- None.

Requirements:

- Subtitle generation should be provided for the following languages as part of the pre-visioned project prototypes and given the provided testing data sets:
  - Finnish
  - Swedish
  - English
  - French
  - Dutch
  - Norwegian (Optionally)
- The service is given a set of spotting rules as input parameters. These spotting rules define, amongst other things these properties (in descending priority):
  - The permitted lines per subtitle;
  - The maximum number of characters per line;
  - The permitted maximum word rate (per minute);
  - The minimum gap between subtitles, and maximum gap allowed between subtitles and the nearest shot-cut boundary;
  - The preferred length of a subtitle as a range of seconds;
  - Linguistic preferences for formatting subtitles (e.g., which words should not be split, transformation of digits into words, etc.).

  Subtitles are generated in such a way that the spotting rules are taken into account to the maximum extent possible. Considering that the input and spotting rules can present contradictory requirements, the output should be optimized such that a maximum score is obtained.
- The spotting rules are provided as input to each subtitle generation invocation because they can change per language and per subtitling organization or per subtitled program genre.
- Output subtitles are represented using the richest format possible, such that new markup can be easily added to the generated subtitles output.

7.12 Content description generation

As a more elaborate form of the video captioning component (cf. 7.7), the content description generation service will take video captions and combine and extend them such that it outputs a human-like narrative to describe the visual characteristics of the video content. As an optional extension, this component will provide these descriptions in a form that can be used to auto-generate audio descriptions for vision-impaired audiences. This epic will be delivered in a collaborative effort between tasks T2.3, T5.4 and T6.2.

Inputs:

- Audio classification (IO4).
- Timed speech transcripts (IO5).
- Video captions (IO14).
- Text with detected and disambiguated named entities (IO10).
- Optionally, Subtitles (IO12).

- Optionally, speaker identification (IO6).
- Optionally, person identification (IO7).

Outputs:

- A list of (audio) content descriptions (IO15).

With additional context:

- If applicable and available, information about the narrative structure of the media (e.g., “this is a documentary using a certain structure”).

Requirements:

- A list of content descriptions is generated which are time-aligned such that it can be identified to which temporal part of the content each description applies.
- Generated content descriptions can combine multiple elements from the input metadata (captions, transcripts, etc.) and “re-write” them in such a way that a better or more informative narrative is obtained which gives readers a better insight into the semantics of the (audio)visual content. To this end, it can employ optionally provided context about the narrative structure of the media content, to steer the narrative style in a more coherent direct related to the source material.
- In addition to producing only preferred natural language narrative descriptions, the service can optionally offer alternatives for elements used in the description, e.g., alternatives to words and concepts used in the ‘default’ description. By providing these alternatives, user interfaces can be constructed to allow for intuitive post-editing of content descriptions.
- Optionally, content description generation can take into account additional program metadata delivered along with the audiovisual content, such as production scripts which contain prepared knowledge about the program and which could give insights into the semantics of the program that are not obviously deducted from the spoken narrative (transcripts) or video images (video captions).
- Optionally, the generation of content descriptions is optimized in such a way that they fit in between, and provide complementary information to existing dialogue and relevant sound effects. This way, they could be used as a rudimentary form of audio content descriptions if rendered to an audio signal using a text-to-speech (TTS) system.

7.13 Content segmentation

Content segmentation aims to segment a source program, described by a variety of metadata (see the list of inputs) into sections that share a topic to a given extent. Examples include segmentation of a news broadcast into items, or different shots into a single scene. This component will be delivered from a collaboration between T3.3 and T6.2.

Inputs:

- Video signal (IO1).
- Timed speech transcripts (IO5).
- Video captions (IO14).
- Text with detected and disambiguated named entities (IO10).

- Optionally, Subtitles (IO12).
- Optionally, speaker identification (IO6).
- Optionally, person identification (IO7).
- Optionally, content descriptions (IO15).

Outputs:

- A list of segments, each of which defined by a time range and text with detected and disambiguated named entities (as in IO10) (IO13).

With additional context:

- Manually added annotations by users.

Requirements:

- A list of segments is output, each of which is defined by a single set of topics. Each segment is timecoded and is defined by text with detected and disambiguated named entities. The actual text and named entities contained in the text should accurately describe the content of the segment, ideally using as little text as possible (but still including sufficient elements to properly describe the segment).
- The service may define multiple layers of segments with differing granularity. E.g., grouping content into a set of topic-based segments, and then a single segment to describe the entire program.
- The service may summarize text using wording that is not taken literally from the input if it conveys the same message as the original text. Similarly, the service may introduce terms to summarize segments as it sees fit, provided that each added term correctly describes the segment or its topics. For example, the service might introduce the word "economics" as a topic to describe a segment, even if that word is never literally mentioned in the input data. Ideally, the topics used to classify a segment should be sourced from commonly used vocabularies, e.g., the IPTC Media Topic NewsCodes[12] taxonomy.
- The development work for this task will be executed in the final project year, and the following is the provisional strategy being considered for a practical implementation of this component:
    - On one level, we will combine both the speech transcripts (IO5) and person identification (IO7) as input to a text topic detection process. This process will output overall classifications of text segments. Using a clustering algorithm, we will attempt to locate those sections of the text that are topic-wise related. This will provide a rough high-level segmentation which is to be refined at 'lower' semantic levels.
    - At the intermediate level, we will use the video captions and content descriptions obtained by automatic and manually curated means to further try to detect related audiovisual segments, e.g., if similar video captions suddenly change of setup or background descriptions.
    - At the lowest level, we will attempt to assemble a segmentation that is less granular than shot cuts (IO8) by using scene detection algorithms to single out those parts that are visually alike across shot boundaries.
    - Using segmentations from these three levels of granularity, we will further attempt to cluster the obtained metadata into relevant segments. We will employ the legacy metadata provided by INA and YLE, and which

[12] Cf. IPTC, "Media Topic" NewsCodes Scheme (Controlled Vocabulary), 2010, Defined at: http://cv.iptc.org/newscodes/mediatopic/.

includes manually curated items segmentation, as a ground truth guide for steering the clustering algorithm in the correct direction. Realistically, we hope to segment items that are clearly delineated in terms of image characteristics and topics. Subtle topic transitions or variations in the subject matter discussed in the media will be much harder to detect.

7.14 Relevant TV moment detection

Relevant TV moments detection will identify those moments within a set of video segments which lead to particular viewer interest, both in the content itself and in accessing content enrichments associated with the program. It will serve as a building block for the story generation epics (cf. Epic 6.6). This component will be developed in T3.2 and T3.3.

Inputs:

- Video signal (IO1)
- A list of segments with disambiguated descriptive metadata (IO13)
- Timed speech transcripts (IO5)
- Video captions (IO14)
- Text with detected and disambiguated named entities (IO10).
- Person identification (IO7)
- Optionally, subtitles (IO12)
- Optionally, speaker identification (IO6)

Outputs:

- A ranked list of segments, taken from the input segments (IO13), but with the addition of:
    - A score (the most important segment gets the highest score).
    - A reason for assigning the score if the segment was ranked as interesting (or non-interesting) for a particular reason (IO16).

With additional context:

- Manually added annotations by users that could point at relevant sections of content.

Requirements:

- To give the service the best change of selecting actually relevant moments, it is provided with a wide set of input metadata, similar to the set provided to the content segmentation component (cf. 7.13).
- To provide systems that use the result of this service a means of evaluating and correcting the relevancy of each selected moment, a reason should be provided for the selection of the moment, e.g., “caused discussion”, “has colorful imagery, etc.

7.15 Spoken language segmentation and classification

Spoken language segmentation and classification segments an audio signal into audio sections labeled with a prediction of which language is spoken in that audio segment. This segmentation will assist downstream processes such as ASR to more efficiently process the audio signal, e.g., by skipping those segments for which no or sub-par support is available.

This service component has been added as a result of a clear need identified during the second evaluation round of the prototype platform, where it was found that items with mixed language spoken content present a significant issue in the auto-enrichment process of audiovisual content (cf. Epic 6.2). In particular, when ASR services (which produce gibberish results for speech that is not ushered in the expected language) are combined with machine translation services (which attempt to translate audio transcripts, regardless of the transcript quality), unmanageable enrichments are created that require a large amount of post-editing to correct. Implementing a language segmentation step and subsequent ASR components which only recognize those sections of audio guaranteed to deliver usable transcription results can help solve this issue.

At the same time, consortium partner AALTO demonstrated[13] significant progress in this area, such that the creation of an integrated language classification component was deemed feasible to implement as part of the final year of the MeMAD project. This service will be developed in T2.1.

Inputs:

- Audio signal (IO2).

Outputs:

- Audio segmentation (IO3).

With additional context:

- Optionally, hints can be provided regarding the spoken languages that can be reasonably expected to be present in the audio signal.

Requirements:

- This service outputs timed segments of audio, each of which is identified with a language label.
- Optionally, a confidence score is provided with each detected segment.

[13] Cf. Matias Lindgren, “Spoken language identification”. Master's Thesis. Aalto University, 2020.

8 Metadata Exchange Format specifications

Using the breakdown of requirements at both individual processing component level (in Section 7), and for the higher-level functional implementation epics (in Section 6), we now define the exchange formats that will be used to exchange data between the MeMAD processing components and the prototype platform.

While the first iteration of the platform was constructed using limited exchanges based on ad-hoc formats, the second and final implementation transition towards a full set of exchanges based on either standardized formats, or new formats that we intent to submit to standardization bodies for extending of existing standards or by defining new best-practices based on the lessons learnt during the research and implementation in MeMAD.

To ensure that the project's results do not overly depend on availability of the Limecraft platform software, we aim to ensure that all underlying components are only loosely coupled to the Flow platform. The core of each component will be able to function independently (and will in many cases also be available as open source software, cf. D2.1), and we will build a suitable architecture where components can easily participate in end users workflows that are orchestrated by the platform's API and that information is exchanged using open (and where possible) standardized metadata exchange formats, as already illustrated in D6.2 and D6.5 (and in the to-be-delivered D6.8). As such, alternative implementations outside of the MeMAD project can also adopt the publicly available components to build their proper processing chains and end-user workflows.

The efforts in constructing the exchange formats for MeMAD described in this deliverable have also influenced the standardization of the EBUCore metadata standard. At the time of writing, this standard was undergoing revisions for its 1.10 release, with suggestions from the MeMAD consortium included. We will report the details of this activity in deliverable D7.3.

8.1 Metadata exchange formats for MeMAD

Given that many metadata exchanged in the MeMAD project serve the purpose of defining descriptive information about audiovisual content resources, we aim to find common ground for the exchange of most data by introducing a metadata framework that can be re-used for many forms of data exchanges.

Based on the metadata modeling work done in T3.1 (of which the conclusions can be found in D3.1), we have defined this metadata exchange framework, which we describe as the **MeMAD Base**, as follows:

1. We use the EBUCore[14] metadata framework for describing the context of the metadata, i.e., for pointing to the information that refers to the audiovisual

[14] "EBU CORE Metadata Set (EBUCore)", Version 1.8, EBU Tech. 3293, 2017, available at: https://tech.ebu.ch/docs/tech/tech3293.pdf .

content being described. Typically, this involves identifying the program that is being annotated, listing its contributors and basic properties (e.g., owner, subjects, but also high-level content descriptions and references to other basic classification properties as provided by Dublin Core, etc.) and defining its relation to the audiovisual media and information systems where the item is stored. Also, support is available for recursively describing segments of metadata (using the same elements as that of the top-level media item), such that editorial 'parts' can be identified and described, which will be essential for describing MeMAD metadata.

2. We use the Web Annotation Data Model[15] for describing actual annotations or metadata that describe the audiovisual content at a low semantic level. Annotations can be made of various types and contents, which we will describe below. The Web Annotation Data Model is supported by the Media Fragments[16] mechanism to enable annotations to refer to segments (temporal, spatial or audio and video track-wise) of the audiovisual content.
3. We incorporate the Annotation model from the NLP Interchange Format (NIF) 2.0 Core ontology[17] (we use the prefix *nif* for these elements later on) for describing the context of auto-generated analytics metadata of various types.

Upon this framework we integrate other specifications or custom extensions (described in the following sections) to model exchange formats for each of the input/output data described in the previous section. Just as is the case with its constituent parts EBUCore and Web Annotations, we define the MeMAD Base and most other metadata models we introduce using the Resource Description Format (RDF). This enables us to easily define an exhaustive ontology of all standardized and custom information objects and their data properties, and our data has the potential to be easily understood by third-party processing systems. Additionally, their serialization can be done in various different forms, including RDF/XML[18] to better align with legacy (semantic web-oriented) systems, but also as the terser JSON-LD[19] for easier integration with recent web development frameworks and REST APIs.

Conversely, for some data such as subtitles, clear and self-contained formats already exist outside the MeMAD Base framework, which we will employ instead of devising new or sub-optimal tweaks our the format.

We next provide an overview of the metadata specifications for the metadata exchanges defined across Sections 6 and 7. The exhaustive definition and examples for each format

[15] Web Annotation Data Model, W3C Recommendation, 23 February 2017, available at: https://www.w3.org/TR/annotation-model/.
[16] Media Fragments URI 1.0 (basic), Version 1.0, W3C Recommendation, 25 September 2012, available at: http://www.w3.org/TR/media-frags/.
[17] Natural Language Processing Interchange Format (NIF) 2.0 Core, introduced in: "Integrating NLP using Linked Data". Sebastian Hellmann, Jens Lehmann, Sören Auer, and Martin Brümmer. 12th International Semantic Web Conference, 21-25 October 2013, Sydney, Australia, (2013); Available at: https://persistence.uni-leipzig.org/nlp2rdf/specification/core.html
[18] RDF 1.1 XML Syntax, W3C Recommendation, 25 February 2014, available at: https://www.w3.org/TR/rdf-syntax-grammar/.
[19] "A JSON-based Serialization for Linked Data", Version 1.0, W3C Recommendation, 16 January 2014, available at: https://www.w3.org/TR/json-ld/.

are available online in a GitHub repository under the MeMAD project organization[20]. Some formats have not yet been fully described due to ongoing modelling and implementation tasks that are yet to be completed at the time of writing of this document. These formats will be finalized as part of the final MeMAD integrated prototype implementation. Examples of these specifications will be updated in the Git repository as progress is made.

8.2 Video (IO1) and Audio Signals (IO2)

We have not explicitly discussed or defined recommended formats for actual audiovisual content in this document. The aim of MeMAD is not to further the state of the art in audiovisual encoding or storage, and hence, MeMAD will use commonly used formats such as the ISO Base File format (MP4), Quicktime (MOV) or Material Exchange Format (MXF) as container formats, and will determine a limited set of commonly supported audio and video codecs to be used for distributing content to and from the project's processing services.

8.3 Speech segmentation information (IO3) and audio classification (IO4)

Speech segmentation information and audio classification both use the MeMAD Base framework, and each segmentation or classification element is composed of two objects.

- The first is the annotation which describes the classification itself and has an RDF class *type* of both a *nif:Annotation* and *memad:SpeechSegmentation* or *memad:AudioClassification* respectively. The *nif:Annotation* is used to indicate that the object is the result of a processing component and the classes from the MeMAD ontology indicate exactly what kind of information is conveyed by the metadata.
  This annotation also has the following distinguishing properties:
  - From the Internationalization Tag Set (ITS)[21] (prefix: *itsrdf*), we use *itsrdf:taClassRef* to identify the detected audio class or speech segment type. We currently employ a list of MeMAD-defined classes for this property (e.g., *memad:MaleSpeakingVoice*), but these can also be sourced from other specifications, e.g. from the AudioSet ontology.
  - *nif:taClassConf* contains the confidence score of the classification according to the processing component, as a decimal value from 0 to 1.
  - *nif:taClassProv* indicates the classification processing component used such that the provenance of the generated metadata can be traced back to its source. This property is typically expressed as the component's software name and version.
  - *rdf:value* represents a textual description of the identified class (i.e., of *itsrdf:taClassRef*) which can be used for quick and user-friendly visualizations of the classification results.

[20] Cf. https://github.com/MeMAD-project/Interchange-formats/tree/master.
[21] "Internationalization Tag Set (ITS) 2.0, W3C Recommendation, 29 October 2013, available at: https://www.w3.org/TR/its20/, and its ontology is defined at: http://www.w3.org/2005/11/its/rdf

- The first annotation is wrapped by a Web Annotations Data Model annotation (of type *oa:Annotation*), for which the first annotation represents the body. This web annotation also contains the media target reference to where in the audiovisual data the annotation applies, expressed as a media fragment URI. Given the exclusive temporal character of the audio signal, the media fragment reference will consist of a time range and optional audio track identification.
  We recommend this web annotation to contain the following extra properties:
  - From the Dublic Core Terms ontology[22] (prefix: dc), we use *dc:creator* to point to an organization or user that initiated the creation of the classification metadata.
  - *dc:created* which indicates the moment in time when the metadata was generated.

  Note that the combined annotation approach allows optimizations in case multiple segments are meant to refer to the same classification (or, e.g., to the same person, cf. also 8.6). In those cases, a single *nif:Annotation* is declared and then used as the body for all applicable Web Annotations (of which each defines its own unique media fragment relation with the underlying audiovisual media).

Examples for audio segmentation and audio classification are provided in the Git repository.

8.4 Timed speech transcripts (IO5).

Many commercial ASR providers provide a proprietary exchange format for ASR transcripts, often tightly coupled with the API service they provide. On the other hand, there is no structured format currently available that serves the requirements defined in Section 7.3. Some ad-hoc formats are used in the research community by toolkits such as Kaldi[23] or the SCTK Toolkit[24], but no prevailing viable formats exist for the purposes of MeMAD.

As part of the development for the second and final MeMAD platform iteration, a format was defined to represent timed speech transcripts. This format was chosen such that it offered maximum compatibility with other MeMAD metadata (i.e., using EBUCore and the Web Annotations Data Model), while also offering the detailed elements (such as per-word timing information) returned by ASR services in MeMAD.

The timed speech transcript format is constructed from the following pillars:

- The format is based around the EBUCore *TextLine* element which is also used by the IO10 (cf. 8.9) and IO11 (cf. 8.10) formats. This element already has standardized support for language and speaker information which can be reused. The TextLine element serves as the top-level element to represent a single speech paragraph or speech included in a single speaker turn. Given that is part of

[22] DCMI Metadata Terms, Dublin Core Metadata Initiative Recommendation, 20 January 2020, available at: https://www.dublincore.org/specifications/dublin-core/dcmi-terms/.
[23] "The Kaldi Speech Recognition Toolkit", Povey et al., Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding, 2011.
[24] "The NIST Scoring Toolkit", National Institute of Standards and Technology (NIST), Version 2.4.11, available at: https://github.com/usnistgov/SCTK, November 2018.

EBUCore, it can seamlessly be embedded into other EBUCore metadata documents (cf. IO19).

- The *TextLine* does lack more detailed per-word timing information and confidence scores that we need for implementing the MeMAD use cases. To support this feature, we devised an object class of *memad:SpeechPart*, which as grouped as a list to the *TextLine* element. Each *memad:Speechpart* contains a limited set of properties:
    - *ebucore:startNormalPlayTime*, the start of the speech part, in milliseconds of normal media playtime from the start of the audiovisual content;
    - *ebucore:duration*, the duration of the speech part, in milliseconds of normal media playtime;
    - *memad:word*, the word detected as spoken in the speech part;
    - *memad:SpeechPartType*, the classification of the speech part (which can be "LEX" or “PUNCT”);
    - *nif:taClassConf*, the confidence score with which the word was chosen for this speech part;
    - *itsrdf:taIdent*, an optional property to identify the speaker associated with the spoken word. Typically, this element can be omitted if the speaker is the same for the entire *ebucore:TextLine*.
- The *ebucore:TextLine* element is given an additional type in the form of a *memad:SpeechParagraph*, to indicate that detailed timing is available for this *TextLine*. The following properties are also defined:
    - *nif:taClassProv*, which can be used to identify the software component that generated the time speech transcript;
    - *ebucore:textLineStartTime* and *ebucore:textLineEndTime*, which convey a summary of timing information at the level of the entire *TextLine* element.
    - *ebucore:textLineContent*, which similarly summarizes the flattened text of the entire *TextLine*. This helps applications not interested in the details of individual word timing data to directly obtain the transcript content without deeper processing.
    - *ebucore:hasTextLineRelatedCharacter*, which identifies the speaker of the speech paragraph, which is an equivalent field to *itsrdf:taIdent* but is used in this context as it is already part of the EBUCore specification.
    - *ebucore:textLineLanguage*, identifies the language of the speech transcript, using RFC 5646 notation based on ISO639-1 language codes.
    - *foaf:gender*, optionally identifies the gender of the speaker.

Regarding serialization, we recommend JSON-LD 1.1 for the serialization of this particular metadata format. This allows for a compact representation that is compatible with light-weight processing systems that have no notion of RDF-based data structures. Using JSON-LD, the timed speech transcript can easily be transmitted with limited overhead and be processed using standard JSON processing libraries. At the same time, it serves as a valid RDF representation and as such can be processed just as other RDF serializations.

Examples for timed speech transcript exchanges are provided in the Git repository.

8.5 Speaker identification (IO6)

The structure of the exchange format for speaker identification largely follows that for audio classification and segmentation (cf. 8.3) and is conveyed using the MeMAD Base and a *nif:Annotation* with the additional *memad:SpeakerIdentification* type. Depending on the richness of the process output, the identification could include only a name, or a complete disambiguation, as follows:

- When the person identification is delivered in a completely disambiguated way, a URI to a descriptive resource is linked using the *itsrdf:taIdentRef* property.
- If only a name is given, then the *itsrdf:taIdent* element is used.

In both cases, the *rdf:value* of the annotation is set to the string label of the identified person, such that a user-friendly label of the identification can always be conveyed. In cases where the identification is provided in an anonymous way (e.g., to represent the match with a nameless voice profile), the *itsrdf:taIdentRef* element refers to the unique identifier of the profile that was matched.
Note that the *taIdent** properties replace the *taClass** properties used for classification in 8.3. Similarly, provenance and confidence score information are here placed in the *nif:taIdentProv* and *nif:taIdentConf* properties. As with IO3 and IO4, the media fragment points to a time range and optional audio track identification of where the person has been identified.

An example of speaker identification is provided in the Git repository.

8.6 Visual person identification (IO7).

In a format almost identical to speaker identification, visually detected persons are conveyed the same mechanism, but in this case with a type of *memad:VisualPersonIdentification* and using both a temporal and spatial media fragment reference. Depending on the richness of the process output, the identification could include only a name, or a complete disambiguation for which the mechanism is identical to that of IO6 (cf. 8.5).

An example of visual person identification through face recognition is provided in the Git repository.

8.7 Shot-cut boundaries (IO8).

Shot-cut boundaries can be trivially described using the MeMAD Base, in which the *type* of the annotation is also set to *ebucore:Shot*. Given the temporal character of the data, the media fragment reference will consist of a time (range) identification. A confidence score and provenance property can be added to the annotation under the form of the NIF *taClassConf* and *taClassProv* properties.

An example of shot-cut boundaries is provided in the Git repository.

8.8 Translated text fragments (IO9).

As with timed transcripts, there are no structured contemporary formats to represent text translation output that conforms to the requirements made in Section 7.8. Many existing translation services provide their own formats, often using JSON including just those features that support the data returned by the service in question.
On the other hand, there has been research in the definition of class models for expressing multi-lingual ontologies, including the Lemon ontology with support for expressing how translated words and concepts relate to one another[27]. While not ready to service machine translation services as is, this previous work can be incorporated for modeling concepts for the translation exchange format.

A collaboration is ongoing between T6.2 and WP4 to define a format which expresses sufficient details of a machine translation to support various downstream processes, including subtitle translation and cross-lingual content retrieval.
This definition will be completed for the final iteration of software components delivered in T4.3 and T4.4. Examples will then also be added to the Git repository.

8.9 Text with detected and disambiguated named entities (IO10).

Named entities can be indicated using a combination of the MeMAD Base and the NIF 2.0 Ontology. The text that is processed and enriched is stored as an EBUCore *TextLine* instance, which is then described using a Web Annotation and a *nif:Annotation* which is also of the *nif:EntityOccurence* class. This NIF *Annotation* contains a disambiguated link to the entity's unique URI, as well as type identifiers for the entity (e.g., sourced from the NERD ontology) and, optionally, disambiguated labels (in one or more languages). Specifically, this information is conveyed in two dimensions as entities are both classified and identified as follows:

- Entities are classified as in IO3 and IO4, using the *itsrdf:taClassRef* property and values from the NERD ontology (e.g., *nerd:Person*).
- Entities are identified as in IO6 and IO7, using the *itsrdf:taIdentRef* property which points to resources describing the entity, e.g., from DBPedia or Wikidata.

With each classification and identification instance, the provenance and confidence data can also be conveyed, as explained in previous sections.

Finally, a provision has been made to allow disambiguated entities to refer to a position in the original text from which they have been detected. For this, we employ more elements from the NIF ontology: *nif:beginIndex* and *nif:endIndex* point to the textual positions within the *ebucore:TextLine* text content.

An extensive example is provided in the Git repository.

[27] "Lexicon Model for Ontologies": W3C Community Report, Cimiano et al. Eds, 10 May 2016, available online at: https://www.w3.org/2016/05/ontolex/.

8.10 Semantically enriched text with detected and disambiguated named entities and links to related resources (IO11).

This metadata extends that of the format IO10 from 8.9, and adds additional Web Annotation instances with references to external resources. These additional annotations can have as target either the audiovisual content, or another annotation or metadata element.

Again, an example of this data exchange format is provided in the Git repository.

8.11 Subtitles (IO12)

Contrary to many other types of metadata to be exchanged in the MeMAD project, the exchange of subtitles is a field for which many formats have already been adopted, throughout the professional media industry (e.g., EBU STL and EBU-TT), the web industry (e.g., WebVTT or TTML) and enthusiasts communities (e.g., SRT). From this variety of formats and standards, we have selected the EBU-TT format for adoption in MeMAD. EBU-TT is a subset of W3C's TTML, optimized for broadcast and web video application use. It supports all required capabilities used in the industry, incl. subtitle positioning, subtitle markup, subtitle coloring, etc. in an extendible format (as it is based on TTML), which is an ideal match with the use cases addressed in MeMAD.

Given the fact that an existing and well-documented standard is proposed for adoption here, many examples can be found from EBU[28] or BBC (from BBC Academy[29] or the Subtitle Guidelines for developers[30]).

8.12 A list of segments with disambiguated descriptive metadata (IO13)

Audio-visual program segments can be represented using EBUCore's *Part* element. When a program is split, EBUCore *Parts* are generated (each with a time delineation with respect to the original content). These *Parts* can in turn be described using all other metadata elements defined in this section, and that of one or more NIF and Web *Annotations* of type IO10 in particular.

An example is provided in the Git repository.

8.13 Natural language video captions (IO14).

The basic form of natural language captions can be expressed using the MeMAD Base, in which the Annotation class is set to *memad:VideoCaption*, and its *rdf:value* is set to the generated video caption. Captions address the video stream using a temporal and

[28] EBU Timed Text Example Files, cf. https://tech.ebu.ch/groups/subtitling#implementations.
[29] BBC Academy: "How do I create subtitles?", cf. http://www.bbc.co.uk/guides/zmgnng8.
[30] BBC Subtitle Guidelines, version 1.1.8, April 2019, cf. https://bbc.github.io/subtitle-guidelines/.

optional spatial media fragment reference, of which one or more can be defined (either for the entire caption, or for a part of the caption, e.g., for describing the position of one of two persons in the image). Apart from this, the same concepts as those from IO3, IO4, IO6 and IO7 are reused.

An example is provided in the Git repository.

8.14 (Audio) Content Descriptions (IO15)

There are two incarnations of content descriptions to be considered. The first are purely textual content descriptions, aligned on the content timeline. These can largely reuse the interchange format defined for natural language video captions (IO14), with potential extensions for editing capabilities for content descriptions, as is currently being considered during the execution of T5.4.

With regards to the extension to describe audio content descriptions, we propose a different approach. As audio descriptions have a close relationship to spoken dialogue and subtitles (which form the counterpart purpose of audio descriptions, but in the visual domain) we aim to express audio descriptions using an extended version of a subtitling format. Such an expression can seamlessly blend both accessibility features into a single delivery document.
In fact, the W3C TTML version 2.0 (working draft)[31] includes a number of provisions for audio synthesis of timed text elements (incl. audio signal gain, panning of audio between channels and speaking and pitch instructions) to support audio description cases[32]. Even though this specification is still ongoing finalization it is an important candidate format for audio description. Its development will be followed up while this topic and the final definition of metadata format will be further defined as part of constructing the content description prototype in T5.4. Examples will then also be added to the Git repository.

8.15 Ranked segments with disambiguated descriptive metadata (IO16)

To represent ranked segments, the format defined for IO13 in Section 8.12 is extended such that each segment EBUCore *Part* is given the required score and score reasoning through a NIF *Annotation*.

An example is provided in the Git repository.

8.16 Production Scripts (IO17)

Very few open reference formats exist for representing production scripts, not in the least because there are many kinds of scripts. In drama production, scripts typically

[31] Cf. Timed Text Markup Language 2 (TTML2) (2nd Edition), W3C Working Draft, 23 June 2019, available at: https://w3c.github.io/ttml2/index.html.
[32] As an implementation of the Media Accessibility User Requirements, W3C Working Group Note, 03 December 2015, available at: https://www.w3.org/TR/media-accessibility-reqs/.

formatted according to a common screenplay format, but more loosely defined formats exist for other types of programs, e.g., spreadsheets or text documents.

Apart from the need to support proprietary formats such as the Final Draft XML screenplay format, we can adopt EBUCore version 1.8+ which has been extended with elements that represent sections of a script, up the level required for MeMAD needs. This includes the *TextLine* element, links from *TextLine* to contributors (actors or roles), references to editorial decisions (e.g., scenes or items to which the *TextLine* belongs), etc.

As far as use cases for this project is concerned, this level of functionality will suffice for supporting production scripts: referring to characters can be done from the TextLine element, and we can link to person elements using IO6 and IO7 exchange metadata. As such, this data can also be exchanged as IO19 (cf. 8.18), which inherently incorporates all of EBUCore as its core components.

8.17 Edit Decision Lists (IO18)

Full-featured edit decision lists are typically conveyed using de facto industry file formats, such as flavors of XML (used by Apple's Final Cut Pro and Adobe Premiere) or Advanced Authoring Format[33] (AAF, used by Avid Media Composer and a variety of other editing tools). Unfortunately, these formats feature only limited extensibility and in some cases are stored in binary form, complicating their use.

Depending on the complexity of the required exchange (does it need to express every single editing decision including effects and the like?) we will adopt the following formats:

- For exchanging simple composition information (including tracks of consecutive clips, without further complexities), the EBUCore *part*s model can be used to express editing information, which aligns with other uses of segments and program sections in this Section.
- More complex EDL representations will use the AAF format, which is commonly used by the industry and can represent the full range of editing decisions used in media production. There will typically be no need to produce new AAF files; existing AAF files will be included as part of the exchange, and they will only be processed to a limited extent required by the representative use case, for example, for performing media tracking.

8.18 Audiovisual program context metadata (IO19)

When exchanging metadata information for a program in its entirety, we will employ the TV program annotation model based on EBUCore to represent this information package, as defined by D3.1 and D3.2. This deliverable demonstrates this use for existing sets of program metadata, but the same concept will be re-used for newly produced content. The EBUCore standard is ideally suited for this. To enable these exchanges, we

[33] "Advanced Authoring Format Object Specification", Version 1.1, AAF Association/AMWA, 2005, at: http://aaf.sourceforge.net/docs/aafObjectModel.pdf.

construct a top-level EBUCore envelope to host the program at the highest level with the metadata that applies to the entire program. This metadata is then added using the formats described above, which is trivial due to the common RDF structure used by most formats. Additionally, metadata that point to program segments are added through EBUCore *Parts*, which can recursively be associated with the same kinds of metadata as for the entire program.

As part of the final delivery of the MeMAD prototype, examples of this context metadata and its inclusion of various other project metadata will be generated from live demonstration data and will then be added to the Git repository as examples for this exchange format.

9 Building the final prototype system implementation

This document provides the required specifications to finish the implementing the MeMAD integrated platform in its final form. As we did with the second iteration, over the course of the following months, the implementation epics defined in Section 6 will now be prioritized and then executed using UCD and agile software development practices[34], in order to obtain a functional final version of the platform that can be evaluated by end users throughout the final year of the project.

In this planning process, the intended evaluation calendar for the final project year, as defined in D6.6, will be taken into account to ensure that components are delivered in time of their evaluation.

Section 10 in D6.6 provides a preliminary outlook on the developments to be done for the final prototype, which remains valid at the time of writing of this deliverable. A detailed and final functional description of implemented features for the final version of the prototype will be reported in D6.8.

[34] Cf. "Agile Software Development with Scrum", Schwaber, Ken and Beedle, Mike, Prentice Hall PTR, 2001.

10 Conclusions

In this deliverable, the third and final of three iterations, we have defined the completed set of requirements for the integrated prototype MeMAD platform. This document defines the functional requirements of the MeMAD prototype system, based on input concerning the tools developed in WP2, WP3, WP4 and WP5, based on the project's use cases from which many requirements are dictated, and finally also based on the evaluation round of the second prototype iteration.

In this deliverable, the previsioned functionality requirements have been grouped into logical implementation epics and the requirements for those and their supporting components delivered by all relevant work packages have been defined. We have also finished the description of the various metadata exchange formats to be used between MeMAD components and the integrated platform. Finally, this document also updates the initial set of evaluation criteria to determine the performance of the prototype system and to help steer the development of media and metadata processing components throughout the project consortium.