Media Fragment Description

Author: L. Nixon – MODUL, Austria, email:lyndon.nixon@modul.ac.at

While media assets could be fragmented in endless arbitrary ways, most use cases would expect that each media fragment is distinctly different in a sense which is meaningful to the end customer. For example, a video frame could be fragmented into spatial regions which each contain a distinct object or a video track could be fragmented into temporal regions which each contain a distinct event. In other words, media is split into different fragments in order to separate out the distinct concepts or topics which it contains. In order to be able to retrieve or re-assemble those fragments, the multimedia system which stores and manages the fragments needs to have access to data about which concept or topic a fragment is representing, which leads to the requirement for *media fragment description*.

Manual description of media assets by experienced annotators has long been the standard approach by media intensive industries (those for whom the appropriate rediscovery of the archived media assets is of particular value, hence justifying the high overhead cost of employing annotation staff). However, as media production and re-use becomes an activity performed by any enterprise, a trade-off is emerging between the higher accuracy of manual annotation and the cheaper production of (semi-)automatic descriptions of the media (based mostly on existing metadata and human input on the one hand, but sometimes by low level feature analysis on the other[1]). The creation of media fragments for each media asset adds a new complexity to the task of annotation, since now descriptions should not only capture the meaning of the whole media asset but the distinct meaning of each fragment of that asset.

This report looks at solutions to the generation of descriptions of media fragments which will enable multimedia systems to retrieve and re-assemble sets of fragments according to the concepts or topics they are representing. This covers the following requirements, where for each we will consider current market solutions, their availability and their maturity:

- Data schemas/models for descriptions of media fragments
- Conceptual vocabularies for the reference to concepts or topics in media fragment descriptions
- Manual annotation tools for the production of media fragment descriptions
- Tools for generating media fragment descriptions from existing metadata and human input / use of conceptual vocabularies in these tools
- Tools for generating media fragment descriptions from low level media feature analysis / use of these tools to support the manual annotation tasks

[1] The techniques which are now applied not only to generate a description of the media asset but also to support the creation of media fragments from it, see the chapter 7.2 in this same document "Media Fragment Creation".

1 Purpose

Enterprises are discovering the value of metadata for their media assets. Search and retrieval of non-textual content –images and videos- could be done through analysis of feature similarity between media but queries are typically not based on similarity to other media but by humans in terms of natural language queries for media representations of specific concepts or topics ("Give me images of ..."). Basing retrieval on feature similarity fails in this regard since measured similarity between features such as audio frequency or image colour map does not necessarily map to conceptual similarity. Hence media metadata has tended to be captured in terms of textual descriptions, which can be matched to user's textual queries. The most basic metadata system could be envisaged as the relationship of some text with individual media assets, with text search retrieving media assets based on term marching within the text descriptions attached to those assets. However, use cases in media search and retrieval have found that:

1. search for media assets may divide the search criteria along different, distinct characteristics of the media itself. For example, the search may wish to restrict results to those created during a certain period, or by a specific author, or from a certain source. Hence media descriptions must capture values of these different characteristics and the media system perform searches applied to them which presupposes knowledge of how those characteristics are identified within the media description. This leads to the definition of **data schema and data models for media description**, such as the Dublin Core property set[2]. While more complex metadata models may make the initial annotation task more effortful, they support richer, more precise searches.
2. differences in how the same term may be specified syntactically within a media description can lead to loss of search precision in media systems. Typical examples are calendar system used ("18.04.13" or "2013/04/18" or "18th April 2013") or how names are entered ("Lyndon Nixon" or "Nixon, L." or "Mr L.J. Nixon"). Such differences in syntax need to be normalised, or at least a means provided to indicate that different syntactic terms in a media description can be considered to be the same concept or topic. This presupposes **a single, referenceable identity for individual concepts or topics** to which different syntactic constructs in a media description can be applied, e.g. through normalisation to an agreed format.
3. use of natural language text in the media descriptions is more 'natural' for the human annotator but can produce problems in search and retrieval for computer systems who are unable to deal with the ambiguities of human language. Expert annotators who draw on an internalised domain vocabulary have been a solution in media intensive industry, where also the searcher is equally an expert and able to formulate their query in the same domain vocabulary. Now that the task of media description and subsequent query is becoming more generalised – being a part of enterprises in any industry domain – these **domain vocabularies need to be externalised and used within annotation tools**, so that produced annotations are tied to controlled terms where possible.
4. automated media descriptions fail to achieve the same level of accuracy as those created manually by human experts. While adding descriptions to media fragments may be less time intensive than seeking to produce a description of a full media asset, manual annotation remains cost intensive especially at growing scale[3]. It is clearly desirable for enterprises sitting on growing archives of media content to shift the annotation effort as much as possible to automated solutions, which in turn increases the importance of **automatic annotation approaches need to become more accurate**, e.g. by being

[2] http://dublincore.org/documents/dcmi-terms/ Dublin Core properties are typically supported in Media Asset Management systems.

[3] Interesting approaches to tackle this include crowdsourcing the annotation work or clustering media assets by similarity so that the annotation applied to one asset is "inherited" by all similar assets.

trained with the enterprises own media assets. As a corollary, manual annotation effort can be reduced by integrating media fragment description into the existing media workflows and tools at the enterprise, extracting as much actionable knowledge as possible from existing human inputs and activities as well as media features and automatically generated metadata, and **reducing human correction of descriptions to the minimum**, e.g. prompting on an ambiguous term for a choice of intended meanings.

The above issues are significant to justify the expense that is generated in introducing richer media fragment description into a media workflow. There is both fixed and variable costs: the initial extension of existing media workflow tools to include the generation of media fragment descriptions is the fixed cost (though subject to technology upgrades and refinements over time too) while the continuous additional resources used to manually correct/check generated descriptions acts as a variable cost. However, while costs can be reduced over time by improving the accuracy of the automated results and reducing the resources needed in human correction of descriptions, as (fragments of) the media assets of the enterprise become more easily retrieved and re-used - both internally and possibly across organisational boundaries – it is the growing benefits, both tangible (e.g. resale of media fragments, lowered in-house media production costs) and intangible (e.g. increased use of media in marketing material or that the re-use of media in user generated remixes help the enterprise become more visible and strengthen its brand[4]), that will drive the investment in richer (stricter data schema, more complex data models), more granular (fragment level) descriptions of organisational media assets.

> Our use case partner VideoLectures.NET) hosts more than 16.000 video lectures from prominent universities and conferences mainly from natural and technical sciences. Most lectures are 1 to 1.5h long linked with slides and enriched with metadata and additional textual contents. Videolectures.NET is being visited by more than 15.000 unique visitors from all over the world daily, which provides a very efficient distribution and dissemination channel.
>
> Visitors typically have limited time to find and watch the materials they want and the topics they search for may be orthogonal to the materials themselves (be the subject of different parts of multiple learning resources rather than the subject of a specific complete learning resource). Visitors would benefit from easier and quicker access to those different parts in the form of a single, integrated presentation of learning materials.
>
> For this, VideoLectures.NET needed to move from atomic descriptions of complete video lectures – powering a search that only found matches on terms in lecture titles and descriptions – to Media Fragment Descriptions of their material, so that user search could discover distinct fragments of different lectures which share the same topic.

[4] Enterprises are coming slowly to the realisation that instead of protecting some of their media assets, it may be more valuable to allow it to be re-used in UGC (user generated content), where an enterprise brand is strengthened by viral distribution of an associated image or video snippet. There is „potential for a stealth campaign to gradually introduce distinctive footage that encourages remixing“ (http://www.hypebot.com/hypebot/2013/08/music-marketing-with-mixbit-exploit-the-oddities-expect-changes.html)

2 Method

Media Fragment Description requires the production of media descriptions at fragment level during and after the media asset's creation and ingestion into the media repository, and management of more structured metadata about each media asset (fragment). Previous reports have covered the steps of Media Fragment Specification and Creation, so we assume that the media assets have been fragmented and that their fragments are also registered in the media repository. There may also be existing metadata which is describing the whole media asset. The methodology required to introduce Media Fragment Description into the media workflow of the enterprise makes use of two distinct infrastructures:

- The **data infrastructure**, mainly back-end (repositories, databases), which is capable to collect, store and expose for search and re-use functionality the media fragment descriptions, including the integration of additional data services which complement the schemas and models used in those descriptions;
- The **tools infrastructure**, mainly front-end (whether user or system-facing), which are capable to gather information about the media, convert it to media fragment description, and supply it to the data infrastructure.

A data infrastructure for Media Fragment Description presupposes a much stricter data schema and much richer data model for the descriptions than lightweight media metadata like Dublin Core. Data generated according to these schema and models will be rendered more precise through the use of externalised domain vocabularies and agreed identity schemes for the concepts the fragments are annotated with. The infrastructure must be capable to enforce (validate) the schema and models and incorporate the foreseen vocabularies and identification schemes, which may be external to the organisation and thus need to be securely integrated as part of the media fragment description workflow and/or cached/replicated internally.

While stricter schematic rules (e.g. date format) could be enforced with specifications like XML Schema, and the resulting XML serialised descriptions stored if desired in specialised XML databases, XML as a data model has not proven adequate for the semantic description (= description of the meaning) of media. For example, different XML documents (in terms of content) can describe the same media resource in the same way – yet differences that have no relevance to the semantic of the media description like the ordering of XML elements can impact on the results of document queries [MM04].

Thus the W3C specification Resource Description Framework (RDF)[5] is preferred as a data model as it is graph based (in its core, a subject resource is connected to an object resource via a property resource, and as each RDF statement is of the form “subject-property-object”, they are often referred to as 'triples') and different syntactic RDF documents (which can use different serialisations, including XML, for storage and document exchange) can still be parsed to the (semantically) same RDF graph. XML Schema can be used within RDF, which is important since object resources (values of properties) may still require syntactic constraints, but RDF introduces as well semantic constraints, i.e. resources can be typed (as anything) and constraints defined for types (in RDF Schema[6]).

For example, one media file could be typed as Slideset and another typed as Video and the schema could enforce that every Video has at most one associated Slideset. Queries can be constructed like “which videos do not have a slideset”, while an attempt to apply two Slidesets to a single video would result in a validation error.

[5] http://www.w3.org/RDF/
[6] http://www.w3.org/TR/rdf-schema/

For using RDF, off the shelf RDF repositories (a.k.a. "triple stores") are available where data can be inserted, queried and manipulated, as well as validated against a provided RDF schema[7]. Fragment descriptions can be stored separately from the media itself, since the reference can be made via media locators in the descriptions which conform to digital identifiers in the media repository. The schema used in the RDF descriptions will need to include the concept type of Media Fragments, and allow associating fragments to the containing media asset (or Media Resource) (and possibly intermediate granularities) as well as annotating them with the concepts they are seen as being representative of. This can be illustrated in a simple graph structure, see Figure 1, which represents the minimal semantic expressiveness of a RDF model for Media Fragment Description.

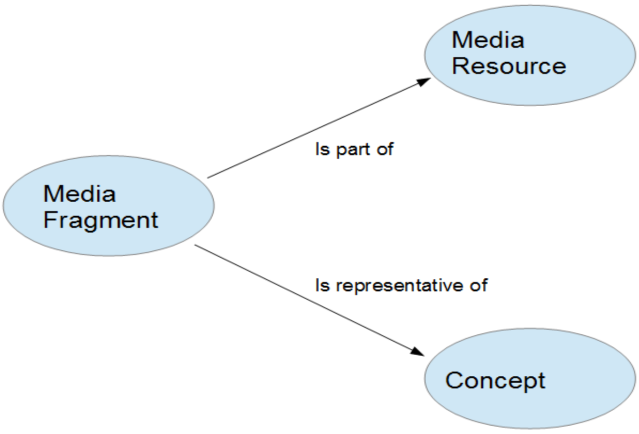


Figure 1: Media Fragment Description RDF model

As a specification intended for integration with the global Web infrastructure, RDF uses URIs (Universal Resource Indicators) as the identifier syntax for concepts in RDF documents. This has the advantage of allowing organisations minting new identities to use their own, already universally unique, domain name as a namespace for the concept URIs. It has also developed as a good practise to make available some human or machine readable documentation at that URI so that any human or machine who encounters that identifier can determine something about what it is intending to represent.

Since typically the annotations of media fragments will make use of concepts which are generally recognised by humans across organisations in their domain, the re-use of domain vocabularies with agreed identifiers for concepts is another advantage of RDF. As soon as the vocabulary is published online and uses URIs to identify concepts, those URIs can be re-used in others' RDF descriptions and their intended meaning proofed by humans or machines by parsing the information published at the URI itself (in HTML for human consumption and RDF for machine consumption). This set of principles around RDF and URIs has become known as Linked Data8.

While more domain specific vocabularies are available in RDF (e.g. GeoNames for geographical information) a commonly re-used vocabulary is the RDF-ization of Wikipedia, known as DBPedia, which provides for every concept which has a Wikipedia article an unique DBPedia URI from which human or machine readable information is available to confirm the intended meaning of the concept, e.g. if we need to refer to "Apple":

[7] https://www.w3.org/2001/sw/wiki/Category:Triple_Store

[8] http://linkeddata.org/ has the definitive list of tutorials and guides to the subject of Linked Data

"The apple is the pomaceous fruit of the apple tree, species Malus domestica in the rose family (Rosaceae)." (http://en.wikipedia.org/wiki/Apple) → http://dbpedia.org/resource/Apple
"Apple Inc., formerly Apple Computer, Inc., is an American multinational corporation headquartered in Cupertino, California" (http://en.wikipedia.org/wiki/Apple_Inc) → http://dbpedia.org/resource/Apple_Inc

The choice of appropriate vocabularies for identification of concepts in annotations is important on several levels. While an internal, proprietary identification scheme could be used, existing tools for describing media assets – as well as other tools in the media fragment workflow – could also support choosing concepts from known online vocabularies such as DBPedia.

Since the vocabularies are available online and use URIs within their own, unique, namespace, they provide a guarantee of shared meaning in the identifier used – also across different applications – where its intended meaning can be easily checked via the URI itself. Finally, these shared, online vocabularies provide more than just an identifier scheme. Concepts are tied to an underlying knowledge model (often called 'ontology'), e.g. all DBPedia concepts are typed and those types are also organised into taxonomic models such that more specific and general types can be found, and concepts of those types identified.

The published information about concepts often includes more metadata about the concept itself, e.g. DBPedia will provide for concepts of type PopulatedPlace values for properties like urban area, population, elevation, long/lat coordinates or the date established. Hence search over the fragment descriptions can be supported by integrating additionally this extra metadata within the RDF store, so that fragments annotated with cities could now be found by queries like "show me places with more than 5 million inhabitants" or "show me videos shot before 1910 of places established after 1900".

Since this approach relies on external, online sources, the data can also typically be replicated within a local data store – also just partially (e.g. only metadata for DBPedia concepts of a certain type) – and synchronized at different intervals if necessary. Another issue for re-use of vocabularies like DBPedia is that the data is not always consistent or correct (it is extracted automatically from the Wikipedia articles), but RDF also allows for the maintenance of an internal, curated RDF vocabulary which can be linked out to external vocabularies like DBPedia, so that curated data can be preferred in the media system but still fall back to external sources when necessary.

The tool infrastructure for Media Fragment Description needs to be both integrated with the existing media systems and with the Linked Data infrastructure discussed above. The automatic capture of relevant information about the media fragment, its conversion to the Media Fragment Description model and storage into that data infrastructure should first and foremost be hidden from the human user, and can be based on the extension of existing systems with components which can access data being passed within the system.

Since this data may take different formats and have differing levels of ambiguity, different wrappers are required for RDF production and will vary from fully automatic (where the original data is well defined and the resulting annotation can be considered to be accurate) to fully manual (where human oversight is necessary to ensure the correct annotation). Such wrappers need to be plugged in too at human data input points, since this is both where ambiguity can arise (e.g. in natural language descriptions or tags) and can be avoided (e.g. in that the user interface prompts the user to use terms from a controlled vocabulary). Tools will disambiguate the determined concepts for annotating the subject media fragment by using URIs from agreed vocabularies – often DBPedia is a default choice for this, model the description in RDF and be able to push the description to a RDF repository.

For VideoLectures.NET it was important not to have to invest in a completely new infrastructure. Instead, the data infrastructure was extended with a metadata repository (Sesame) to store the RDF descriptions of the video lecture fragments. This complements the existing text indexes of the video lectures and the media asset repository where the videos are stored with associated data (e.g. transcripts).

The RDF production is handled by a combination of several analysis services. A script on the media asset repository could call these services with references to the respective data for analysis when a new media asset is added in VideoLectures.

3 Tools

Media Fragment Descriptions can be generated by any tool capable of processing a suitable input (either the media asset (metadata) itself or an UI to a human who knows the media asset (metadata)) and generate from it an appropriate RDF description which could be saved to a RDF repository. However here we do not, for example, look at generic RDF creation tools, which require an understanding of the chosen RDF data model and semantics[9]. Rather, our interest is in a tool set which can allow companies with media assets (which have already been fragmented) to produce accurate semantic descriptions of their media fragments at minimal effort and cost. There are two types of tool to consider here:

- those which can be integrated within existing media systems, wrapping internal data flows;
- those which allow media systems user to check, correct and complete fragment descriptions.

1. **Wrapping internal data flows**

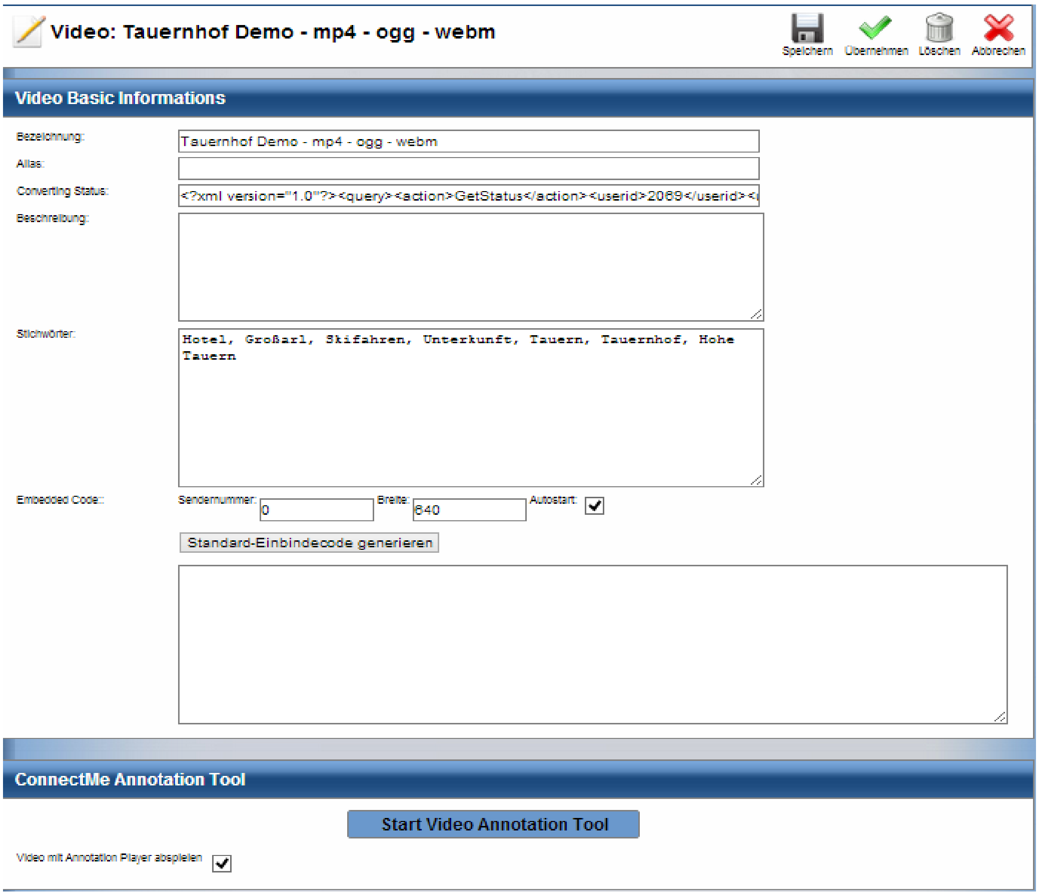
While the means of association of metadata with media in a media workflow can be workflow specific, there are two types of metadata creation and association, which generally occur in acquiring and managing media assets. Media at the point of creation may already store some metadata associated with it. If an organisation is not the creator of the media, then they are dependent on the creation metadata which is provided, e.g. a typical example of this is the camera settings, date/time or GPS information stored by digital cameras with photos, which can then be seen again in photo software or online photo sites[10]. Another set of metadata may be generated at the time of ingestion of media assets into a repository, e.g. a human editor may add title, description, keywords etc. At scale, it is likely media assets are loaded automatically into repositories with little to no additional human annotation, and possibly only at the point of selection and intended use that a human editor will add some useable metadata to the asset to aid its future retrieval or further re-use. As such, metadata for these types of tools may be limited, yet the leveraging of existing metadata to create initial fragment descriptions is always beneficial, to reduce the remaining cost and effort needed by a human annotator when checking and completing descriptions in specialised annotation tools.

Metadata mappings from a CMS

[9] e.g. https://www.w3.org/2001/sw/wiki/Category:RDF_Generator lists different tools and APIs which extract RDF from a Website or database structure. These require initial configuration with a mapping from the underlying structure to the RDF vocabulary.

[10] A short overview can be found at http://graphicssoft.about.com/od/glossary/f/metadata.htm

In the ConnectME project (www.connectme.at) media fragment descriptions were generated out of industry partners existing media Content Management Systems (CMS): in one case, a proprietary CMS capable of exporting the human-input video metadata in the mediaRSS format; in the other, a Drupal CMS used as media repository extended by a specialised RDF export module for the media metadata. The below screenshot (Figure 2) shows the extended proprietary CMS, where the legacy metadata fields filled in by the media channel owner (title, description, keywords for example) are complemented by a "Start Video Annotation Tool" button.



Video: Tauernhof Demo - mp4 - ogg - webm

Video Basic Informations

Bezeichnung: Tauernhof Demo - mp4 - ogg - webm

Alias:

Converting Status: <?xml version="1.0"?><query><action>GetStatus</action><userid>2069</userid><

Beschreibung:

Stichwörter: Hotel, Großarl, Skifahren, Unterkunft, Tauern, Tauernhof, Hohe Tauern

Embedded Code:: Sendernummer: 0 Breite: 640 Autostart: ☑

ConnectMe Annotation Tool

Video mit Annotation Player abspielen ☑

Figure 2: Media description integrated into CMS (courtesy Yoovis GmbH, yoovis.at)

When the button is pressed, the metadata in the proprietary CMS for this media asset is published to an API on the metadata repository using the mediaRSS format. An internal script on the repository side maps the mediaRSS information into RDF metadata (see below). In the case of Drupal, a dedicated RDF Module[11] is used to write Drupal node data into a RDF model and then a dedicated Publishing Module[12] for the metadata repository is able to publish this RDF to Linked Media Framework, which is being used as media metadata repository.

Which CMS?

Drupal is currently to be preferred as media CMS due to the support for RDF which is built in to the software since Drupal 7. The core module outputs a version of RDF that can be embedded inside the HTML webpage (RDFa), there are separate tools to extract the RDFa and convert it into

[11] https://drupal.org/project/rdfx
[12] https://code.google.com/p/lmf/wiki/DrupalModule

another RDF serialisation which can also be added to the Drupal installation. There are also means to add REST support to Drupal so that e.g. RDF can be exported via HTTP to another application. It fits well with metadata repositories which support RESTful endpoints.

Other CMS may only support differing data models for their data export, necessitating an additional wrapper to be implemented on either the CMS (if extendible) or repository side (or even in-between) to map this into the RDF model for the repository.

Which metadata repository?

The choice of repository is free to the implementer, all support different types of APIs for importing RDF from another application but we suggest support for REST should be preferred. This is a simple Web-friendly specification of Web service interfaces using HTTP, so that the data exported from the CMS can be (HTTP) POSTed to the repositories REST Web interface.

Linked Media Framework (LMF) was preferred due to its support for handling media metadata – while metadata stores generally just contain references to the content being described (e.g. via an URL for online content or some other DOI for content in a media asset management system) LMF allows for the media content itself to be stored together with the metadata and supports content negotiation on the media content URL to allow applications to request either the media itself or its metadata using only the known URL (something its developers have called part of the 'Linked Media principles'[13])

The Linked Data repository part of LMF is now being further developed by Apache as Marmotta (http://marmotta.apache.org/), which will include support for queries over Media Fragment Descriptions which recognise Media Fragment URIs (SPARQL-MM), e.g. ask for fragments before, after or contained in other fragments[14].

In both cases, the mappings make use of two relevant means to generate an initial media (fragment) description, and as such can serve as a template for any organisations development of mappings for the internal media data in their systems. As seen in the examples above, further decisions to consider are whether mappings can be performed within the media management system or metadata repository or must be “wrapped” by an external service/script – modularly extendible CMS like Drupal or metadata repositories like Linked Media Framework help ensure this runs well integrated to existing data flows – as well as how the various systems can communicate with one another – we benefit from the use of media repositories and metadata repositories with flexible APIs for external applications supporting both data input and output.

(a) *Define mappings to property-values.* Original media data often contains information of the form “property-value”, regardless of in which data model and schema this is stored (database tables, CSV, XML etc.), e.g. image dimensions or video duration. These can be mapped directly onto properties of the RDF schema for the media fragment description, whereas the property value may need some transformation to be consistent in the new RDF model. However, since RDF does explicitly model “property-value” relations in its data model and data processing benefits from a consistent usage of property value syntax, this conversion to a normalised RDF format is highly beneficial for the data re-use, in whatever context. Usually such property mappings can only

[13] https://code.google.com/p/lmf/wiki/PrinciplesLinkedMedia
[14] See the Sample Queries at http://demos.mico-project.eu/sparql-mm/sparql-mm/demo/index.html

apply to the media asset as a whole since original media data generally only recognises the whole asset.

Mapping tables for the most common media data schema can be found at
http://www.w3.org/TR/mediaont-10/

Mappings could be implemented in any scripting language able to access the original media data in its provided schema (DB, CSV, XML etc.), and output property-value pairs (while a RDF API for writing to a RDF model or directly publishing to the RDF store is obviously best[15]). Typically some data type manipulation is needed so this should be determined in advance that the scripting language can handle this requirement efficiently. Such a mapping script can be installed just after the insertion of media data into a system so that an initial media description is written to the media metadata store.

(b) *Extract entities for the media fragments.* Some of the available media metadata may be usable for describing the media at the fragment level but this mapping is slightly more complex than defining property-value pairs. Usually titles, descriptions, keywords and so on have as value longer strings where the direct mapping to RDF does not add any new information about the media. However from those strings entities may be identified that can be considered 'relevant' to the media item. By identifying entities using URIs from Linked Data vocabularies, media systems can benefit from the additional information about those entities that is available and machine-processable via those URIs. So instead of just having a video whose title is “Obama meets Merkel at the UN”, we could have a RDF description also annotating the video with the concepts of Barack Obama, Angela Merkel and the United Nations. Now a video search can use concept meaning in finding a video rather than just string matching , e.g. a query like “videos where US Presidents and German Chancellors meet” can now be answered by using the concept typing (which is given in the Linked Data) to match this to the concepts of Obama and Merkel. To perform this entity extraction, external services can be called with the string as input and return the found entities. Of benefit to the media fragment description, there are some file inputs where temporal information is also attached to text fragments, so that an entity extraction service for media fragments could additionally link the found entities to a temporal segment of the media. Typically this is found in subtitle files (e.g. SRT format) and video captioning (e.g. WebVTT). Since most entity extraction services only offer a link between the extracted entity and the positioning of the (sub)string which refers to that entity in the text file, for those services an additional processing step is needed to infer from that positioning information the temporal fragment in which the string occurs. Note that while it is less typical to refer to spatial information when describing an image, the presence of such in image metadata could allow the same association of extracted entities and media spatial fragments.

Where do I find an entity extraction service?

Different entity extraction services exist, with differing terms of use, licenses, language support and supported input/output formats. It is worthwhile to test a few with typical media metadata from your organisation to determine which offer the best accuracy for extracting relevant entities for your content, taking into the account the types of entity you want to extract. This paper [GAN13] provides an overview of entity extraction services whose output uses Linked Data URIs for the identified entities. An alternative is to install and use a software package which performs entity recognition, which may be preferable if you need to train the service to recognise entities drawn from a specialised vocabulary (public services tend to look at DBPedia entities and other general collections, e.g. placenames from a geographical dictionary). Apache Stanbol

[15] https://www.w3.org/2001/sw/wiki/Category:API

(http://stanbol.apache.org/) can be installed as a Web application, or as a module in the Linked Media Framework, and used for entity extraction with one's own dictionary of terms.

The NERD service (http://nerd.eurecom.fr/api) and Web interface (http://nerd.eurecom.fr) aggregates results from many different online entity extraction services, allowing users to both maximise results or to compare between them. NERD also supports subtitles and video captions as an input, returning both entities and their temporal fragment in the media [LI12].

LinkedTV Metadata generator

In the LinkedTV project (www.linkedtv.eu) a Web service is provided for ingesting various types of related information for a media asset and generating RDF descriptions as a result. The significant contribution of this service is that not only does not accept several different types of input but it outputs the full Media Fragment Description in RDF, aggregating the results of all the different input processing steps:

- Exmaralda, a format for aggregating media analysis results obtained after the execution of different low level feature analysis processes over media content. They include shot segmentation, scene segmentation, concept detection, automatic speech recognition, between others. (this is the output format of LinkedTV analysis services, see our report on Media Fragment Creation)
- TV Anytime, a metadata format for legacy information from broadcasters. This is an example of the “mappings to property-values” discussed above.
- SRT subtitles file, using entity extraction and associating the entities to a temporal fragment of the media, as discussed above. This step uses NERD, and the NERD output can also be directly input to the service if preferred.

A Web interface for the metadata generator is available at http://linkedtv.eurecom.fr/tv2rdf/.

A Web API will be made available subject to similar terms of use as NERD in Summer 2014.

2. **Manual tools to check and complete descriptions**

Annotation tools provide an interface to a human annotator to check the existing, automatically extracted descriptions (from the above tools), correct errors and complete the descriptions such that they are ready for actionable usage within the media system for asset retrieval and re-use. A key issue with current semantic annotation tools is the lack of agreement on the data models that they support to import into the tool and to export from the tool (e.g. which metadata schema is used for properties of the media, which vocabularies are used in the identification of the concepts in the annotation such as Linked Data). Furthermore, many tools are tied to a specific metadata repository for both reading and writing of the annotations. We have recently surveyed how open, flexible and ready for media fragments and Linked Data the current bunch of “semantic multimedia annotation tools” are:

- Annomation[16]: this is a browser based tool for video annotation. It is currently restricted to educational material available within the Open University. Tags can be added at any point in the video timeline and given a duration. A number of vocabularies are supported for the tags, including DBPedia and

[16] http://annomation.open.ac.uk/annomation

GeoNames. The resulting video annotations re-use several ontologies, but seem to be saved back into the tool's own repository, i.e. they are only available again to the same tool.

- Annotorius[17]: this is an image annotation tool which is browser-based, implemented in JavaScript. It allows the attachment of free text descriptions to a spatial region. A Semantic Tagging plugin suggests named entities for the inserted text, which map to DBPedia resources. Annotations use their own JavaScript data objects for persistence and sharing.
- YUMA[18]: developed in the EuropeanaConnect project, it supports image, audio and video. Both DBPedia and Geonames resources can be annotation targets, and are suggested from free text or location references respectively. Annotations can be exported as RDF using a tool-specific vocabulary.
- SMAT[19]: Semantic Multimedia Annotation Tool - promises to allow the annotation using domain ontologies of fragments of content items within a rich internet application. Video can be accessed from any streaming server and annotated with spatial or temporal fragments connected to a term from a preloaded domain ontology. It is targeted to pedagogical usage and seems to be focused on demonstrating the act of media annotation rather than any wider re-use.
- SemTube[20] is a prototype for semantically annotating YouTube videos developed within the SemLib EU project. It allows attaching annotations to both spatial and temporal fragments, with annotations being free text, Freebase terms or full RDF triples. A faceted browser then allows users to explore their annotated videos. It appears functional but seems to be only enabled to save and retrieve annotations within a host server.
- Pundit [21] is an open source Web document annotation tool that has developed out of the SemLib EU project. It incorporates however only image annotation at the moment, allowing regions of the image to be annotated with LOD terms or freely chosen ontology URIs. A client can be downloaded and installed for local annotation of online Web pages which are saved to and retrieved from an instance of a Pundit server.
- IMAS[22] is a Web-based annotation tool developed within the SALERO EU project. Structured descriptions can be produced for media assets retrieved from a repository. SALERO developed its own ontologies for annotating media and describing relationships between media according to the needs of the media production domain. The tool only allows global annotation of media resources and not annotating parts of them, and the output is specifically intended for the needs of producers (e.g. subsequent rediscovery of media) rather than for publication to the Web.
- ImageSnippets [23] enables to tag images using Linked Data resources. Interestingly, tagged images can then be published to the Web, both with descriptions embedded in the image data and included in the HTML as RDFa metadata, on the fly. However, the tool does not yet support fragment-based annotation and it is restricted to the image medium. It is currently in beta but looks promising, except that its current open annotation approach could suffer from shared public image annotations not being interoperable due to a lack of a common annotation vocabulary among authors.
- OpenVideoAnnotation[24] plans to offer a web-based tool to collaboratively annotate video on the web, at the fragment level and using the Open Annotation ontology. Annotations are free text comment and tag

[17] http://annotorious.github.io/

[18] http://dme.ait.ac.at/annotation/

[19] http://www.kp-lab.org/tools/semantic-multimedia-annotation-tool-smat

[20] http://www.semedia.dibet.univpm.it/semtube/

[21] http://thepund.it

[22] http://salero.joanneum.at/imas/

[23] http://www.imagesnippets.com/

[24] http://www.openvideoannotation.org/

based, but it is not yet clear if Linked Data will be supported nor if spatial fragments will be included. This tool is clearly promising but this is still a work in progress with a soon to be launched beta program.

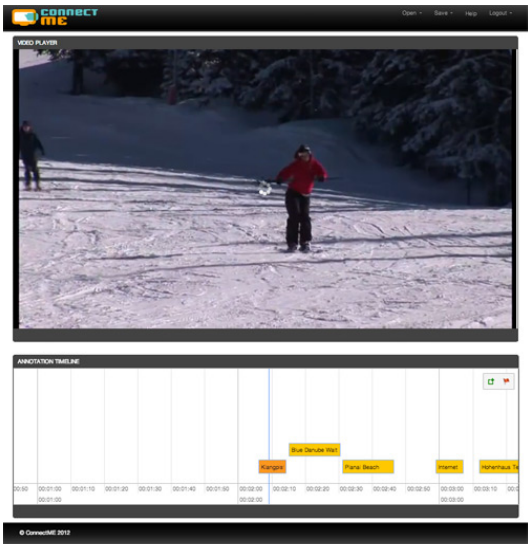
Having reviewed the most recent tools known to the authors, we highlight work of two projects the author has been involved in which are continuing the task of supporting online semantic media annotation with Media Fragment and Linked Data support.

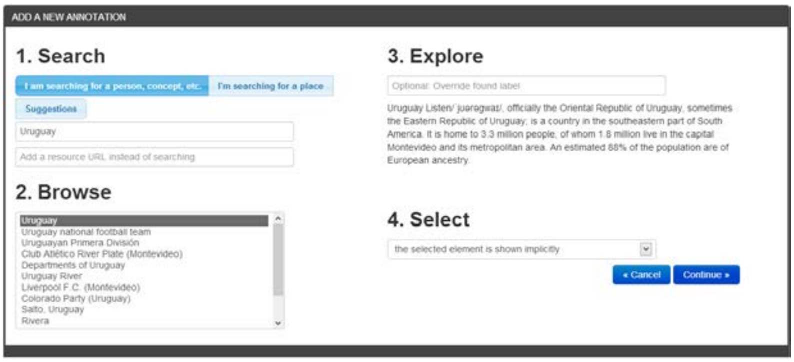
ConnectME video annotation tool

In the ConnectME project (www.connectme.at) a Web based annotation tool has been developed. It is part of the workflow discussed above where the initial media mappings are published to an instance of the Linked Media Framework. In the Web tool, a media asset may be selected and its annotation inspected, using a timeline view (Figure 18 left) under the video frame to clearly show descriptions along the media's temporal fragments and allow editors intuitive editing of temporal fragment start/end times by drag & drop. Spatial fragments are also displayed for a selected annotation, if present, and can be changed by drag & drop of a spatial overlay over the video frame. The annotations are shown with their concept labels, with the addition/editing of annotations taking place in an easy-to-use window (Figure 18 right) which allows plain text entry and suggests concepts to the annotator, providing a preview text to allow checking the correct concept is selected.

The ConnectME annotation tool can be configured to connect to any Linked Media Framework instance, offers videos it finds in the repository for annotation (video formats have to be supported by the HTML5 implementation of the Web browser to be played back. MPEG-4, WebM and OGG Video are the preferred formats.) and allows the annotation that has been made to be downloaded to the user's machine in RDF or saved back into the metadata repository.

It is currently available for download under CC-BY-NC-ND license from https://git.sti2.org/projects/CONNECTME/repos/annotation-tool/browse[25], derivative and commercial use conditions on request. A test installation can be used at http://annotator.connectme.at/




ADD A NEW ANNOTATION

1. Search

I am searching for a person, concept, etc. | I'm searching for a place

Suggestions

Uruguay

Add a resource URL instead of searching

2. Browse

Uruguay
Uruguay national football team
Uruguayan Primera División
Club Atlético River Plate (Montevideo)
Departments of Uruguay
Uruguay River
Liverpool F.C. (Montevideo)
Colorado Party (Uruguay)
Salto, Uruguay
Rivera

3. Explore

Optional: Override found label

Uruguay Listen/ˈjuərəgwaɪ/, officially the Oriental Republic of Uruguay, sometimes the Eastern Republic of Uruguay; is a country in the southeastern part of South America. It is home to 3.3 million people, of whom 1.8 million live in the capital Montevideo and its metropolitan area. An estimated 88% of the population are of European ancestry.

4. Select

the selected element is shown implicitly

« Cancel Continue »

Figure 3: ConnectME video annotation tool

[25] An open source release via a public code hosting site like GitHub is planned for Summer 2014.

LinkedTV Editor Tool

In the LinkedTV project (www.linkedtv.eu) a Web based editor tool has been developed. It loads the RDF descriptions from the LinkedTV Platform, making a distinction between the media 'entities' (the concepts each fragment is annotated with) and 'enrichments' (in LinkedTV, on the basis of the media annotations, hyperlinks to related content are also suggested). Of course, the Editor Tool can be used to edit only the media annotations. The LinkedTV metadata generator, discussed above, can publish its generated RDF to the LinkedTV Platform so that the descriptions of the media are available to the Editor Tool. The tool allows an editor to select a specific chapter of the video, browse the existing annotations on the platform, and select them for the “media fragment description” or add new annotations. The manually selected annotations are saved back to the platform using a separate graph, which has the effect that – while the existing annotations are preserved and can be returned to – the manually selected annotations can also be selected out from the repository easily. The idea of the Editor Tool is to support content editors at TV broadcasters who will want to proof all automatically generated annotations and use, in the subsequent media workflow, only annotations they manually selected. The Figure 4 shows the main interface to the Editor Tool with the existing platform annotations listed on the right hand side and editor's accepted annotations on the left below the video.

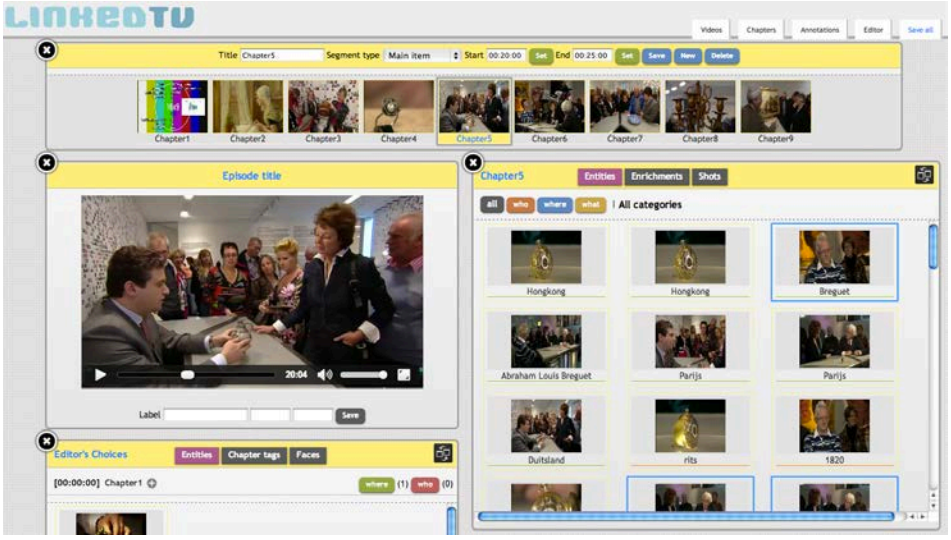

Title Chapter5 Segment type Main item Start 00:20:00 End 00:25:00
[00:00:00] Chapter1

Figure 4: Editor Tool Interface

The LinkedTV Editor Tool will have a public code release in late 2014, but a test installation can be used at http://editortool.linkedtv.eu
It is currently tied to an instance of the LinkedTV Platform but could be reconfigured to read and write RDF from a different repository. It reads and writes RDF according to the LinkedTV Ontology.

For VideoLectures, the base asset metadata in the media asset repository is kept apart from the richer fragments descriptions in the new metadata repository. The latter is used in the portal search functionality to find matching fragments, and via the shared identifiers used for the videos themselves, the fragments are associated with the media assets in the media asset repository.

The TV2RDF service was used to convert both results of shot segmentation analysis and acquired talk transcripts (using, in cases, automatic speech recognition software) into the LinkedTV RDF format.

In the future, a customised version of the LinkedTV Editor could be used to check and correct media fragment annotations, or to allow curated associations between media fragments and other Web content (enrichments) which could extend the browsing offer of VideoLectures.

4 Models/Formats

For a correct functioning of a tool chain for Media Fragment Description the metadata input-output should of course conform to the same data model and schema. There is still some variation in media metadata schema that could be chosen, but in the interest of ensuring a base compatibility across heterogeneous metadata – since many media properties are consistently present in most or all of those schema – the W3C has developed a “media ontology” which provides for a mapping of shared properties across schema[26].

The W3C media ontology has been described in more detail in MediaMixer D1.1.2 “Core Technology Set” along with additional background to the state of the art in multimedia description formats[27].

However, since this model is driven by ensuring consistency across as many different metadata schema as possible what it defines in the end is a 'lowest common denominator'. It does not specifically address any specific requirements of media fragment description such as enforcing the use of the Media Fragment URI specification in identifying media, or natively working with annotations which use Web based URIs for concept identifiers. Such “media fragment description” friendly metadata schema extend, for convenience, existing schema such as W3C Media Ontology such that they could support these requirements, enforcing (with validation) the use of fragmentation in media description and Web based URIs to identify the concepts in those descriptions. Within the context of the above tools, two ontologies (data models for media description with a formal logic basis) are relevant, both of which extend the media ontology and include support for media fragments and for semantic concepts:

ConnectME ontology, used in the ConnectME framework metadata mappings and annotation tool, is a lightweight model (re-using only a subset of the W3C Media Ontology and the Open Annotation Model, extended with a small set of additional properties) which has the media resource in the centre, and annotates on its fragments – the only concepts associated to the resource itself are “subjects” (entities extracted from the media title, description etc. which cannot be automatically associated with a fragment e.g. these are used in the

[26] http://www.w3.org/TR/mediaont-10/
[27] http://community.mediamixer.eu/documents/coretechset

annotation tool to provide a suggestions list when annotating). The fragments are associated to concepts with specific properties (which can be extended) such as *explicitlyMentions* or *implicitlySeen*, allowing media systems processing these descriptions to make distinctions based on the concept is being 'represented' by the media fragment. The ConnectME ontology, see **Figure 20**, is published at http://connectme.at/ontology

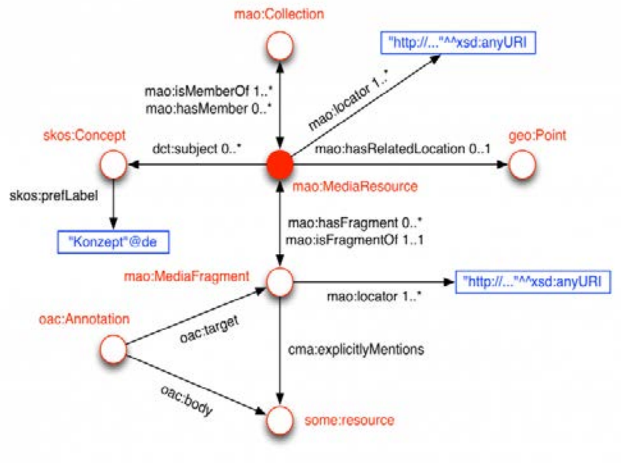


Figure 5: The ConnectMe Ontology

LinkedTV ontology, used in the LinkedTV metadata generator and editor tool, combines several ontologies with a specific extension for its use case of modelling the (automated) annotation of media fragments and their association to related content (hyperlinking). Going beyond the ConnectME ontology, it does also use the idea of MediaFragments annotated (via the Open Annotation Model) with semantic concepts but can also model a wider range of information, including initial outputs from media analysis processes (which can subsequently be used for the semantic annotation), provenance information and different levels of granularity (not just fragments but also Shots, Scenes or Chapters). The effect is that of enabling a fuller media fragment description which fits better the full media lifecycle within a media management process, e.g. able to preserve information about where an annotation came from, which analysis results (e.g. entity extraction from subtitles) were used to create an annotation, or a series of edits to a fragment description via an annotation tool. A fuller consideration of the LinkedTV ontology, see **Figure 6**, is available in the project deliverable D2.4, Chapter 2 [JR13] and the specification is published at http://linkedtv.eu/ontology

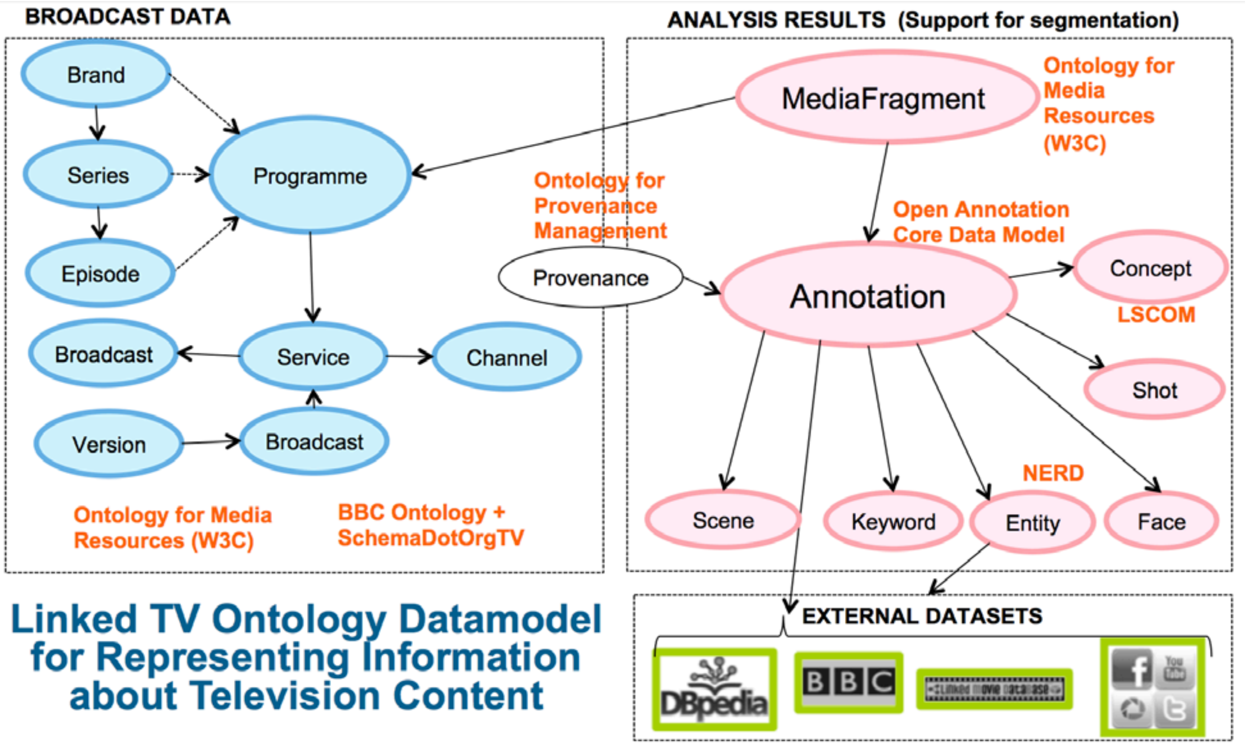

BROADCAST DATA
ANALYSIS RESULTS (Support for segmentation)
Brand
Series
Episode
Programme
Broadcast
Service
Channel
Version
Broadcast
Ontology for Media Resources (W3C)
BBC Ontology + SchemaDotOrgTV
MediaFragment
Ontology for Media Resources (W3C)
Ontology for Provenance Management
Provenance
Open Annotation Core Data Model
Annotation
Concept
LSCOM
Shot
NERD
Scene
Keyword
Entity
Face
EXTERNAL DATASETS
DBpedia
BBC
Linked Movie Database
Linked TV Ontology Datamodel for Representing Information about Television Content

Figure 6: The LinkedTV Ontology

5 Licensing & contact

For more information and access to the described annotation tools, please refer to the table below.

Tool	Link (URL)	Contact person	License
Apache Marmotta	http://marmotta.apache.org	John Pereira Redlink.co john.pereira@redlink.co	Apache 2.0
Apache Stanbol	http://stanbol.apache.org	John Pereira Redlink.co john.pereira@redlink.co	Apache 2.0
Drupal RDF export module	https://drupal.org/project/rdfx	Stéphane Corlosquet Lin Clark	GNU GPL 2.0
Apache Marmotta RDF import module	https://code.google.com/p/lmf/wiki/DrupalModule (current version for Linked Media Framework)	John Pereira Redlink.co john.pereira@redlink.co	Apache 2.0
NERD (Named Entity Recognition and Disambiguation)	http://nerd.eurecom.fr	Raphael Troncy EURECOM r.troncy@eurecom.fr	See http://nerd.eurecom.fr/terms
TV2RDF REST service	http://linkedtv.eurecom.fr/tv2rdf/	Raphael Troncy EURECOM r.troncy@eurecom.fr	Planned: same terms as NERD (above).
ConnectME annotation tool	https://git.sti2.org/projects/CONNECTME	Lyndon Nixon MODUL University lyndon.nixon@modul.ac.at	CC-BY-NC-ND (Planned open source release in Summer 2014)
LinkedTV Editor Tool	http://editortool.linkedtv.eu	Jaap Blom Sound and Vision jblom@beeldengeluid.nl	GNU GPL 3.0 (planned, code release late 2014)

6 References

[GAN13] Aldo Gangemi, “A comparison of knowledge extraction tools for the Semantic Web”, ESWC 2013

[JR13] Jose Luis Redondo Garcia and Raphael Troncy, “Annotation and retrieval module of media fragments”, LinkedTV deliverable D2.4 published at http://www.linkedtv.eu/wordpress/wp-content/uploads/2013/12/LinkedTV_D2.4.pdf

[LI12] Yunjia Li et al. “Creating enriched YouTube Media Fragments With NERD Using Timed Text”, ISWC 2012

[MM04] Jacco van Ossenbruggen, Frank Nack, Lynda Hardman, "That Obscure Object of Desire: Multimedia Metadata on the Web, Part 1," IEEE Multimedia, vol. 11, no. 4, pp. 38-48, October/December 2004