Synote: Weaving Media Fragments and Linked Data

Yunjia Li,Mike Wald,Tope Omitola,Nigel Shadbolt and Gary Wills
School of Electronics and Computer Science
University of Southampton
UK
{yl2,mw,tobo,nrs,gbw}@ecs.soton.ac.uk

ABSTRACT

While end users could easily share and tag the multimedia resources online, the searching and reusing of the inside content of multimedia, such as a certain area within an image or a ten minutes segment within a one-hour video, is still difficult. Linked data is a promising way to interlink media fragments with other resources. Many applications in Web 2.0 have generated large amount of external annotations linked to media fragments. In this paper, we use Synote as the target application to discuss how media fragments could be published together with external annotations following linked data principles. Our design solves the dereferencing, describing and interlinking methods problems in interlinking multimedia. We also implement a model to let Google index media fragments which improves media fragments' online presence. The evaluation shows that our design can successfully publish media fragments and annotations for both semantic Web agents and traditional search engines. Publishing media fragments using the design we describe in this paper will lead to better indexing of multimedia resources and their consequent findability.

Categories and Subject Descriptors

I.7.2 [**Document Preparation**]: Hypertext/hypermedia, Multi/mixed media, Standards; H.5.m [**Information Systems**]: Miscellaneous

General Terms

Theory, Design

Keywords

annotation,linked data,media fragment,schema.org,search

1. INTRODUCTION

The term "media fragment" refers to the inside content of multimedia objects, such as a certain area within an image, or a ten minutes segment within a one-hour video. With the rapid development of multimedia applications, such as YouTube and Flickr, end users could easily upload, share and tag the multimedia resources online. Compared with traditional annotations, semantic annotations can tell a computer how media fragments are related to other data and how these relationships can be processed automatically. However, most applications are yet to expose sufficient semantic annotations for media fragments, which leads to the difficulty of processing complex searching and providing accurate searching results. To solve this problem, some research, such as Core Ontology of Multimedia (COMM) [3], has enabled semantic annotations for media fragments. But this approach has not yet proved to be scalable on the Web [13]. Linked data [6] is more light-weight compared with traditional semantic Web technologies and is highly scalable on the Web. There are some research interests in applying linked data principles to media fragments [13, 21]. The recent announced schema.org[1] also defined vocabularies to improve the online presence of image, audio and video objects, but the media fragment is undefined in schema.org.

Most research about media fragments focuses on exposing the closed data, such as tracks and video segments, within the multimedia file or the host server using semantic Web and linked data technologies. But these efforts are inadequate for the fully interlinking and indexing of media fragments, because the large amount user-generated annotations in existing Web 2.0 applications are rarely published, such as YouTube interactive transcript[2] and photo tagging functions in Facebook[3]. When a person is tagged in a photo, an annotation is created and connected to the media fragment, but the annotation is saved in an external source. More attention should be given to how these external annotations from outside of multimedia host servers could benefit the media fragment interlinking and search.

In this paper, we will present the design and evaluation of an example of publishing media fragments and annotations following linked data principles. We also discuss how the media fragments could be made available for both semantic indexing services, such as Sindice [40], and Google Search. The external annotation data is from Synote [22], which is a Web 2.0 application allowing users to embed audio-visual resources from other domains and make synchronised annotations. The discussion in this paper focuses on the temporal dimension of audio-visual resources, which could also be extended to images and spatial dimension.

In the reminder of this paper, Section 2 reviews related vocabularies that could be used for media fragment publishing, and the state of art of the existing applications about multimedia and linked data. Section 3 briefly introduces the Synote system and discusses the requirements of me-

LDOW2012, April 16, 2012, Lyon, France.

[1] http://schema.org
[2] http://goo.gl/t1nMj
[3] http://www.facebook.com/help/photos/tag-photos

dia fragment publishing. Section 4 discusses the design to solve the problems in interlinking multimedia (iM) principles [13]. Then we present the implementation of the design and a model to allow Google indexing media fragments. Section 5 presents the evaluation results and Section 6 provides conclusions and recommendations for future work.

2. RELATED WORK

Linked data describes a series of methods of publishing structured data using semantic Web technologies, such as Resource Description Framework (RDF) [24] and SPARQL query language [27]. Linked data enables machines to automatically discover more data from the data they already know. Generally, there are four rules which must be followed when publishing linked data on the Web [6]:1) Use URIs as names for things 2) Use HTTP URIs so that people can look up those names 3) When someone looks up a URI, provide useful information 4) Include links to other URIs, so that they can discover more things.

Heath and Bizer [14] summarised that there are some common steps which developers need to take from choosing URIs to the final testing and the publishing of linked data fall into several patterns. They also pointed out that developers should not totally abandon "existing data management system and business applications", but add an "extra technical layer of glue to connect these into the Web of Data".

2.1 Multimedia Annotation Vocabularies

If we want to publish media fragments in Synote, we need to choose how different dimensions are encoded in URIs. We also need to consider which vocabularies should be used to describe multimedia resources and annotations. Thus this section reviews related standards and vocabularies.

The standardisation of media fragments is identified as an urgent need [37]. Many standards use non-URI based mechanisms to identify media fragments, such as MPEG-7 [25] and Synchronized Multimedia Integration Language (SMIL) [17]. In these standards, the descriptions of temporal and spatial dimensions are divided into several attributes, thus the media fragment is not represented by a single URI. In the URI-based approaches, temporal, spatial and other dimensions of media fragments are encoded in query ("?"), hash ("#") or after slash ("/"). The temporalURI [26] defines media fragments after URI query. But Query parameter has a weakness that the resulting URI loses the direct link to the parent resource and it does not follow the "Cool URIs" principle [30], so it is difficult to be applied in linked data. The proposed URIs for Linked Stream Data [31] applied slash URIs to include real time and space information in URIs together or separately. MPEG-21 [8] specifies a normative URI fragment syntax to address fragment in MPEG compatible files. But the syntax of MPEG-21 can be ambiguous and it is difficult to fully follow the standard [23].

The W3C Media Fragment Working Group has proposed Media Fragment URI 1.0 (MFURI) [38], which defines hash URI syntax to address media fragments from temporal, spatial, track and named dimensions. MFURI is included in Ontology for Media Resource 1.0 (OMR) [29], which is an ontology to describe multimedia resource. Hausenblas et al. have discussed the issues about iM using MFURI specification, such as media fragment dereferencing, legacy metadata alignment and interlinking methods [13].

Multimedia Metadata Ontology (M3O) [28] provides a framework to describe and annotate complex multimedia resources. It fills the gap between the structured metadata models, such as SMIL and EXIF [35], and semantic annotations. M3O can be integrated with various standards to provide semantic annotations. The key concept of M3O is separation of information objects and information realizations. Based on this idea, M3O comes up with a core set of ontology design patterns, including annotation, decomposition, collection and provenance patterns.

International Press Telecommunication Council (IPTC) designed an ontology for IPTC News Architecture [36] in order to integrate news and multimedia metadata from the industrial area into the existing knowledge on the semantic Web. In [36], a local view of a video resource is introduced so that arbitrary sequences of a video, i.e. media fragments, can be played in the semantic browser. As the further development of IPTC News Architecture, rNews vocabulary has be adopted by schema.org, which provides shared vocabularies in the form of HTML Microdata [15]. A great advantage of using schema.org is that the structured data including the audio and video objects could be recognised and highlighted by major search engines, such as Google, Bing and Yahoo!. The vocabularies in schema.org can also be used in the form of RDFa[4] [1].

2.2 Multimedia Applications for Linked Data

Many applications have already published multimedia and annotation as linked data, which offers experience for us on multimedia resource publishing. Yovisto open academic video search platform publishes its database containing video and annotations as linked data using MPEG-7, COMM to describe multimedia data [42]. Annomation [20] is a tool to handle the input annotations from users developed by the Open University in the UK. The annotations are saved in RDF quad store with users' own privacy and provenance data. SemWebVid [33] can automatically generate RDF description by analysing plain-text resources using multiple Natural Language Processing (NLP) Web services. The entities extracted from the NLP services are used to generate the RDF description of the video resources.

LEMO multimedia annotation framework [12] publishes media fragments using MPEG-21. LEMO converts existing video files to MPEG compatible versions and streams them from LEMO server. LEMO also derived a core annotation schema from Annotea Annotation Schema [19] in order to link annotations to media fragments identifications. NinSuna (Metadata-driven media adaptation and delivery) [9] has a server-side implementation of MFURI 1.0. The system can "ingest" different formats of multimedia resource and save the fragments information as semantically represented data blocks in RDF. When delivering the media resource, one or more data blocks can be returned to the user agent (UA) according to the Range header in HTTP request as well as the adaptive context of the UA. Even though some of the applications introduced above publish user-generated content with media fragments, most of them are built from scratch in a linked-data-driven method. Our work on Synote in this paper focuses more on publishing media fragments and annotations based on typical Web 2.0 applications.

3. PUBLISHING REQUIREMENTS

[4] http://goo.gl/Zd8Eo

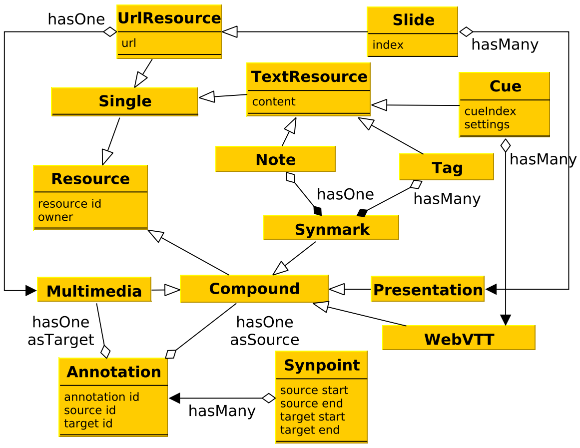


Figure 1: Synote Object Model

Figure 2: Screenshot of Synote Player

This section will briefly introduce the Synote system and discuss the requirements of media fragment publishing for Synote. This section sets up the background for Section 4.

Synote[5] is a Web-based multimedia annotation application [22], which synchronises user-generated tags, notes, transcripts and images with a time-span of multimedia resources (i.e. media fragments). Synote does not have its own multimedia repository and Synote's database only saves the URLs of audio-visual resources and images freely available online. User-generated annotations and the synchronisation points (Synpoint) are saved in the Synote database. Figure 1 demonstrates the backend object model of Synote[6].

Synote categorises the resources into single and compound resources. Single resources are either URI references or text. Compound resources are further divided into Multimedia, Synchronised Bookmark (Synmark, compromised by one text Note and many Tag resources), Transcript (in WebVTT [16] format) and Presentation resource[7]. Each compound resource is aggregated by one or more single resource. Every resource, no matter single or compound, has a unique id in Synote. Multimedia and Slide resources have URI references to online resources, such as YouTube video, audio files and images, etc. Annotation class models the annotation relationship between two compound resources. In Synote, only Multimedia is the target resource and Synmark, Presentation, Transcript could be the source in an annotation. In an annotation, Synpoints associate each single resource in Synmark, Transcript and Presentation with a time-span ("target start" and "target end") of the Multimedia.

Synote Player is the main page to display all these resources and annotations in an interactive and synchronised manner. Users could click on a Synmark, transcript block or slide (part 2, 3 and 4 respectively in Figure 2) and the multimedia player (identified as 1 in Figure 2) will jump to the corresponding start time. In Synote Player, users can also create Synmarks, Transcript and Presentation slides, and synchronise them with the multimedia (Figure 3). Technically, the Synote system follows the Model-View-Controller framework. Synote Player is developed using jQuery[8], so it contains dynamically generated content by javascript (Ajax).

The general principles for iM have been introduced in [13] and most of them are still the requirements for Synote. However, as Synote is not the host of multimedia resources, it is not Synote's responsibility, for example, to deploy the legacy multimedia metadata and we can only publish the metadata saved in Synote's database. For the audio-visual presentation of media fragments, we can still use the interactive annotation highlighting function in Synote Player's page. So the audio-visual presentation of media fragments is actually implemented in the HTML page.

Except for the general requirements of iM, Synote also has its own publishing requirements. Similar to major multimedia applications on the Web, Synote has ready built its own resource management system. So it is unwise to totally abandon the old infrastructure and it is better that the linked data could act like an extra layer in order to minimise the changes to the existing system. This requirement also means we are not abandoning Synote Player, which could be used as the HTML representation of different resources. Unlike a company's registration data, media fragments and annotations in Synote could be generated frequently by users, so it is better to publish data dynamically instead of dumping data to RDF periodically like DBpedia [4]. Since the original goal of Synote is to improve the indexing of PART of multimedia resource [22], we need to consider how to make major search engines index the media fragments defined in Synote. In addition, Synote has a permission system where users can create private resources and they are only accessible after the owner signs in. So the media fragments and annotations about private resources should not be included in the published datasets. Except for Multimedia resource and media fragments, we also need to make other resources and annotations dereferencable, because they are all linked to media fragments and should be published together.

To sum up, there are several requirements we need to fulfil

[5]`http://www.synote.org`

[6]Classes and relationships not related to this paper are not presented in this figure

[7]We capitalise the first character of the resource name to indicate that it is a class defined in Synote Object Model

[8]`http://jQuery.com`

when publishing media fragments. Even though some of them are special requirements for Synote, they could also be concerns of major multimedia applications when publishing media fragments and annotations.

4. DESIGN AND IMPLEMENTATION

This section discusses the design solutions to satisfy the publishing requirements of iM and other requirements mentioned in Section 3.

4.1 Interlinking Multimedia

The ultimate goal of iM is to *"derive both semantic and audio-visual representations from multimedia resources on the Web"* [13]. In this section, we will discuss how Synote is designed to solve the key issues in iM.

MFURI is designed to deal with the closed character of multimedia resources on the Web. It expects the cooperation of "smart servers" and proxy caches to deliver media fragments via HTTP protocol [23]. Hausenblas et al. [13] proposed a solution, which uses "HTTP 206 Partial Content", content negotiation and 303 redirect to retrieve the original and RDF representation of a multimedia resource. There are arguments about the appropriateness of this solution under the Web Architecture [18] because *"a description of a multimedia resource is not the same as the resource itself"* [13]. As far as we are concerned, the third principle of linked data only requires a description of the URI and it is not necessary, as well as not possible sometimes, to provide the semantic description which could totally reflect the other representation of this URI. Even though some UAs, such as Firefox 10[9], and servers, such as Ninsuna, support part of the functions defined in MFURI, the general support for this solution is very limited currently on either client or server. Another solution mentioned in [13] is to embed semantic metadata in the multimedia file like XMP [2], which means the two representations share the same URI. Whichever solution is chosen, the server, which hosts the multimedia resources, is responsible to provide both audio-visual and RDF representations.

Multimedia resources could be hosted in one domain (like YouTube), but embedded in some HTML document (also called "replay page" or "landing page") in another domain, such as wiki or blogs. So many user-generated annotations are saved separately from the servers which host the multimedia files. In YouTube, for example, title, keywords, comments and interactive transcript are displayed on the same page, i.e. the replay page, as the video content. The replay page also controls the audio-visual representation of the media fragments via the embedded player to provide the interactive transcript function. MFURI specification also proposes that the embedded player in the replay page should implement the retrieval functions of "smart user agents".

From the discussion above, we can conclude that the semantic annotations of media fragments are mainly coming from two types of sources. **Type One**: The applications, which offers multimedia delivery service, publishes media fragments with metadata they have. The server has to make sure the media fragments are dereferencable using methods such as content negotiation, embedded semantic annotations with HTTP streaming, etc, and they could be reused by other applications. **Type Two**: The applications, which embed multimedia resources, reuse the media fragments and interlink them with their own annotations. Annotations in Type Two are mainly generated by users and they reflect users' interests in different areas. Applications providing Type Two annotations are actually interlinking their annotations to the media fragments in another domain, but they have no control of what could not be dereferenced or retrieved via these media fragment URIs.

We must emphasise that the two sources are not necessarily in different domains. For example, YouTube has its own video repository but it also has a replay page which embeds the video and allows users making annotations. The data published in Type One and Two can be about the same property of a multimedia resource. An mp4 file can embed the title information using MPEG-7, but when the mp4 file is played in another application, it may be given another title. This is acceptable because vocabularies describing multimedia resources, such as OMR, do not require that all the metadata come from the file itself. For example, the subtitles in OMR could be an external link of the subtitle file or a track media fragment embedded in the file[10].

The annotations in Synote belong to Type Two as Synote does not host any video or image resources. In the following subsections, we discuss how Synote provides audio-visual and semantic representations of multimedia resources, which is also useful for other multimedia applications.

4.1.1 Dereferencing Media Fragment URIs

It is not a good practice to augment the URIs in another domain and include it in your own dataset (see Section 4.1.2.1 of [14]). Synote has no control of the hosts of multimedia resources, so we need to mint our own URIs for each multimedia resource and fragment. In addition, we need to think about how to include both HTML and RDF representations of media fragments as well as other resources in Synote. We already have the replay page, which is *http://synote/recording/replay/id*[11]. The replay page displays all resources and annotations related to a multimedia resource (Figure 2), so we use it as the HTML representation for all resources and annotations, including media fragments as explained in Section 3.

The URI of the replay page *replay/id* is not intuitive to indicate the RDF representation of a multimedia resource. As mentioned in Section 3, we want each resource (Multimedia, Synmark, Tag resource, etc in Figure 1) and annotation to have a unique URI. In the semantic description of a multimedia resource, URIs of other resources in Synote could be included and they have their own semantic descriptions as well. We use *resources/id* to indicate the ids of resources in Synote as non-information resources, and *resources/data/id* to indicate the RDF representation of each resource. Then we use 303 See Other to redirect the request to different documents based on content negotiation [30].

Even though we mint our own URIs for multimedia resources hosted in other domains, the media fragments are still parts of the parent resource. So it is appropriate to use hash fragment to specify the media fragments, instead of using query or slash URIs mentioned in Section 2.1. For Synote, we attach the fragment string in MFURI syntax at the back of resource id URIs, such as *resources/36513#t=3,7* (the resource with id 36513 must be a multimedia resource

[9] https://bugzilla.mozilla.org/show_bug.cgi?id=648595

[10] http://www.w3.org/TR/mediaont-10/#example2

[11] we use *replay/id* for short

in Synote). The content negotiation still works for media fragment URIs. If semantic representation is requested, the request will be redirected to *resources/data/36513* and an RDF file containing all the media fragments and annotations about resource 36513 is returned. If the HTML representation is requested, the request will be redirected to *replay/36513#t=3,7*. Then the fragment #t=3,7 will be attached at the back of the real URI of the multimedia resource that is to be played, for example *example2/1.mp4*. Even though the multimedia host server may not support retrieving the media fragment *example2/1.mp4#t=3,7* and the native player in the browser cannot highlight the media fragments, Synote Player can control the embedded player to start playing from 3s to 7s using javascript and the corresponding annotations are highlighted straight away. In this way, the audio-visual representation of the media fragment is derived based on the HTML representation. One problem about this solution is that different browsers have their own fragment precedence during redirects[12] and some of them do not attach the fragment in request URI to the redirected URI. Then the replay page will not be able to highlight the media fragment on opening the page.

Listing 1: Description for media fragment

```
@prefix ma:<http://www.w3.org/ns/ma-ont#>.
@prefix <#> .
@prefix dc:<http://purl.org/dc/elements/1.1/>.
@prefix dbpedia:<http://dbpedia.org/resource/>.
@prefix oac:<http://www.openannotation.org/ns/>.
@prefix lode:<http://linkedevents.org/ontology/>.
@prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#> .
<> a ma:MediaResource;
   a oac:Target;
   ma:hasFragment :t=1,14;
   rdfs:seeAlso <replay/36513>;
   rdfs:isDefinedBy <resources/data/36513>;
   dc:title "Tim Berners-Lee:The next Web of open,linked data";
   rdfs:label "Tim Berners-Lee:The next Web of open,linked data";
   ma:hasSubtitling <resources/36535>;
   ma:locator <http://youtu.be/abc>.
:t=1,14 a ma:MediaFragment;
   a oac:Target;
   ma:hasKeyword <resources/36634>;
   ma:hasKeyword <resources/36635>;
   lode:illustrate _:event1;
   ma:description "This talk took place at Terrace Theater";
   ma:locator <http://youtu.be/abc#t=1,14>;
   ma:isFragmentOf <resources/36513>;
   ma:title "TED talk";
   rdfs:seeAlso <replay/36513#t=1,14>;
   rdfs:isDefinedBy <resources/data/36513>.
_:event1 a lode:Event;
   rdfs:seeAlso <tim_berners_lee_on_the_next_web.html>;
   lode:involvedAgent <http://dbpedia.org/resource/Tim_Berners-Lee">;
   lode:atPlace <http://dbpedia.org/resource/Terrace_Theater>.
```

Listing 1 presents a N3 format of a sample semantic description[13] for *resources/36513* and its media fragments. We use rdfs:isDefinedBy and rdfs:seeAlso to connect the HTML and RDF representations as suggested in [30]. OMR is used to describe the multimedia resource and media fragments. In this example, ma:locator is used to indicate the real location of the multimedia file or the service which delivers the file. In OMR, MediaFragment is a subclass of MediaResource, so ma:location is also an attribute of media fragment. We simply attach the MFURI fragment string at the back of the location URI, which indicates the real media fragment that should be delivered from the host server.

As we discussed earlier, the media fragment retrieval function proposed in MFURI has not been widely implemented on either client or server side, so we cannot expect YouTube provides semantic description for *youtu.be/abc#t=1,14*[14]. However, including this URI in the semantic description of media fragment is still safe. According to the discussion of httpRange-14 issue[15], as long as the server returns 200 or the 303 redirect can lead us to a 200 response, the URI is considered dereferencable. So if a semantic agent tries to dereference *youtu.be/abc#t=1,14* or any other multimedia files with hash fragment attached, the server can at least return 200 because *youtu.be/abc* does exist. Maybe the semantic description is not provided by the server, but it will not return 4XX or 5XX errors, which means nothing can be determined about the URI. This is also another reason we choose hash URIs to denote media fragment. We can attach query ?t=1,3 or slash /t=1,3 at the back of the multimedia file *example.com/1.ogv* to indicate the media fragment. But we cannot control whether the query and slash is valid on the server and it is quite likely that 4XX or 5XX error will be returned. The hash fragment will not be passed to the server in HTTP header of the request, unless the smart agent embeds it in HTTP Range header by implementing the proposed functions in MFURI specification [23].

This solution gets around the problem that most multimedia host and user agents have not implemented the proposed functions in MFURI specification, but we still want to publish media fragments with annotations generated elsewhere. This solution is still compatible with the multimedia host applications if the functions in MFURI are implemented later.

4.1.2 Describing Resources and Annotations

As mentioned in Section 4.1.1, each resource in Synote is given a URI and the semantic description about it could be dereferenced. In Listing 1, *resources/36636* and *resources/36635* are Tag resource and *resources/36535* is the Transcript resource in Figure 1. Each annotation object defined in Figure 1 is also give an unique URI. We use Open Annotation Collaborative (OAC) vocabulary to model the general annotation relationship. As can be seen in Listing 1, the multimedia resource is the the oac:Target in an annotation instance. The oac:Body could be Synmark, Transcript or Presentation resource which annotates this multimedia. OAC only defines the annotation relationship in a higher level, so more specific properties are also applied in the semantic descriptions. For example, we use ma:hasSubtitling to represent that a Transcript resource is the subtitle of a multimedia resource. Another example is, if a Synmark resource annotates a media fragment, we can use ma:hasKeyword to model the relationship between the Tag resource in the Synmark and the media fragment.

In Synote, each compound resource has one or more sin-

[12] http://trac.tools.ietf.org/wg/httpbis/trac/ticket/43
[13] Some parts of URIs are ignored for short

[14] youtu.be/abc is an HTML document and the actual video content embedded in this document is protected by tokens. YouTube does not give a public URI to its video content, and we suppose that the token can be mapped to the video content. So we use youtu.be/abc as the indirect reference to the video file.
[15] http://goo.gl/XC821

gle resources. This relationship is modelled using Open Archives Initiative Object Reuse and Exchange (OAI-ORE) vocabulary[16]. The following example means *resources/36633*, which is a Synmark resource, aggregates *resources/36634*, which is a Tag resource.

```
<resources/36634> ore:isAggregatedBy <resources/36633>.
```

We also consider the permission control over private resources. If the requested resource or annotation is private or does not exist, our server will return 404 Not Found. We could use 403 Forbidden for private resources, but we do not want to expose to the client that these resources are private (see Section 10.4.4 of [11]).

There is a problem to align the legacy multimedia metadata in Type One with the vocabulary we use to describe multimedia resources [13]. This problem is not critical to Synote because we reuse the vocabularies, which have been accepted by major communities, and we are not creating a vocabulary ourselves which no one else uses. Besides, we have seen some major progress in OMR, which provides the mapping between OMR and other vocabularies.

4.1.3 *Interlinking Methods*

Synote references the URIs of images and audio-visual resources from another domain, and they are published with other resources and annotations in Synote. This could be perceived as "interlinking". However, the real value of publishing media fragments is that they could be reused in different context, especially some domain specific areas. The media fragments published in Type One are usually linked to low level metadata about the multimedia resource, such as framerate, subtitling, etc. To enrich media fragments with domain specific knowledge, we need to think about how the user-generated annotations could be further linked to other datasets in the linked data cloud.

We have been inspired by the work of RDFaCE[17], which is a RDFa Content Editor based on rich text editor TinyMCE[18]. In RDFaCE, users could manually enrich the text by adding RDFa to the content in a user-friendly manner. RDFaCE can also connect to external semantic indexing and annotation services, such as Sindice [40] and Open Calais[19], to automatically generate RDFa for the content. In Synote, when a Synmark is created in Synote Player, rich text or HTML could be added into the Note resource (Figure 3). So we can allow users to manually add or automatically generate RDFa for the Note resource in Synmark. The content of Note resource will be connected to the media fragment using ma:description property as shown in Listing 1. But before publishing the data, we can extract the triples from RDFa and add them in to the RDF file as the semantic description of the media fragment.

In Listing 1, some triples have been added to the Note resource as RDFa to describe the event of the talking using Linking Open Descriptions of Events (LODE) [32]. The lode:involvedAgent of this media fragment is Tim Berners Lee and it took place at Terrace Theater. Then the fragment "#t=1,14" is further linked to Tim and Terrace Theater. As the location of the media fragment is *youtu.be/abc#t=1,14*,

[16]http://www.openarchives.org/ore/
[17]http://aksw.org/Projects/RDFaCE
[18]http://www.tinymce.com/
[19]http://www.opencalais.com/

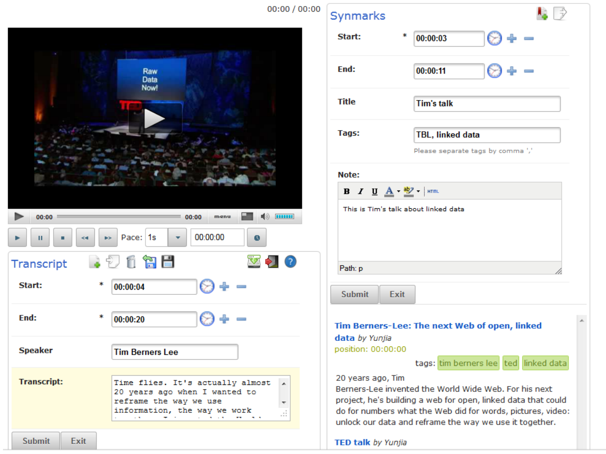


Figure 3: Edit Synmark and Transcript in Synote Player

so this media fragment from YouTube is also linked to Tim and Terrace Theater. In addition, once the Synmark Note resource with RDFa has been saved into Synote's database, this change can be immediately spotted in the dereferenced semantic description for media fragments.

Ninsuna has proposed a way to embed temporal metadata represented by Media Fragment URIs and its description in RDF format in a WebVTT file[20]. Synote also uses WebVTT as the format for the transcript, so we can use this method in a similar way as RDFa editor in Synmark. When users add RDFa or pure RDF triples in the cue body of a WebVTT file, we can extract the triples and add them in the semantic description of media fragments. Furthermore, this solution in the future could be used as embedding structured data in the tracks of multimedia files and the objects will become searchable by specialised search engines [41].

4.1.4 *Summary*

The sources of annotations for media fragments can be divided into two categories. The multimedia server can embed annotations in the multimedia file, or save them separately on the server but delivered using Web standards, such as HTTP streaming. The multimedia sharing applications focus more on user-generated annotations, which are saved externally to the multimedia file and multimedia host server. It has been stated in [38] that "*enabling the addressing of media fragments ultimately creates a means to attach annotations to media fragments*". The "annotation" here should refer to the annotations from both sources and both of them are important. But Synote only has external annotations attached to media fragments, so our design mainly focuses on the annotations in Type Two.

Our design solves the key issues of iM when deploying Synote into the linked data cloud. We totally reuse existing vocabularies to describe resources and annotations, and we did not develop any vocabulary of our own. The current vocabularies are enough for us to publish our resources and annotations, and make sure that all necessary information are included. To describe the annotations about multimedia resources and media fragments, it is better to use more

[20]http://ninsuna.elis.ugent.be/node/39

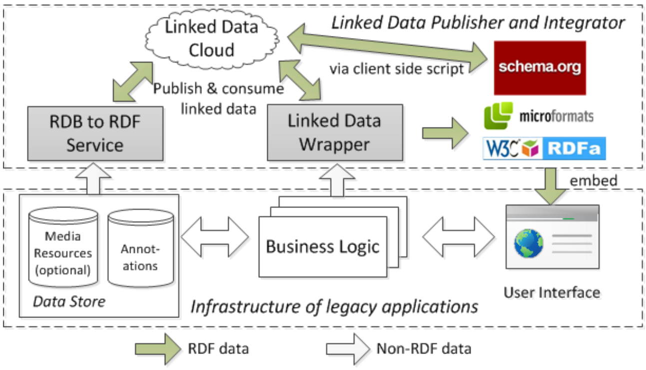


Figure 4: Publishing Patterns Used in Synote

specific properties instead of general annotation vocabulary, because the semantic agents can clearly know different aspects of media fragments and thus more efficiently tailor the search and reasoning results.

We try to make the media fragment publishing compatible with current user experience. Users in Synote can still create Synmarks and transcripts without knowing that the data has been published with media fragments and interlinked with other datasets. However, the interlinking methods are still not very user-friendly as the knowledge about RDFa or semantic Web are required to create Synmark Note. Some (semi-)automated methods of interlinking generation need to be considered in the future work.

4.2 Choosing Publishing Patterns

This section will introduce how the design of iM in the previous section is implemented by realising several linked data publishing patterns.

We want to build an extra layer on top of the existing Synote, so that non-RDF data could be published as RDF data. But migrating the data store from the current relational database to RDF store would result in the major redeveloping of the whole system. Periodically dumping RDF triples from the database and serving them via a Web server could work, but we want to publish the data in real-time. So the patterns left for Synote are RDB-to-RDF service, Linked Data Wrapper and Rich Snippets [34]. Many tools are designed for RDB-to-RDF service, such as D2R server [7], OpenLink Virtuoso [10] and Triplify [5]. These tools often require developers to provide a mapping between database schema and the vocabularies. RDB-to-RDF service only shares the database from the existing application, thus we do not need to change Synote at all. But implementing access control over RDB-to-RDF service is not straight forward. Besides, some functions are difficult to be implemented such as the extraction of RDFa for interlinking (Section 4.1.3).

Our decision is implementing a RESTful API, i.e. Linked Data Wrapper, with only HTTP GET method for the dereferencing of RDF representation and we also embed Microdata in the original Synote Player page using vocabularies defined in schema.org to improve the online presence of the published resources (Figure 4). On the one hand, we embed the URIs, which could be dereferenced through the API, into the Synote Player page using "itemid" attribute in Microdata. On the other hand, we use the existing vocabularies in schema.org and the semantic search engine on indexing the Synote Player page can also find more information through "itemid". This design can also make full use of the existing permission control system in Synote as the private resource will not be displayed in Synote Player nor included in the RDF. As the RESTful API generates RDF at runtime from database, user-generated annotations will be immediately published. The information of start and end time in the media fragment string is saved in Synpoint (Figure 1), so they could be easily retrieved from the business logic of Synote to build the fragment string in MFURI syntax.

The problem of this design is that no SPARQL endpoint is provided by Linked Data Wrapper and Rich Snippet, so it is difficult to execute complex queries over the datasets. That is why we are still expecting to include RDB-to-RDF service as a complement pattern for Synote. Another possible solution is writing a RESTful API. Users can send the query data through the API and the server returns the semantic description as the response. Another problem is we only created several demo recordings with media fragments as a proof of concept, but it is still unclear if this solution is efficient for large datasets with millions of media fragments.

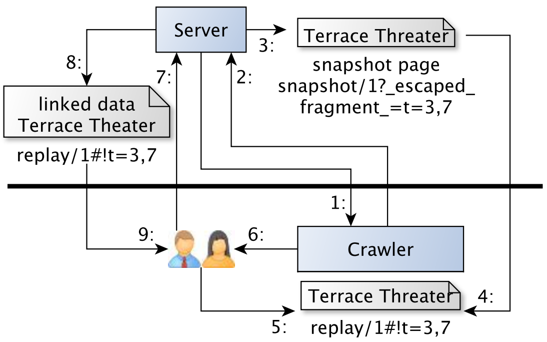
4.3 Online Presence for Media Fragments

The ultimate goal of this work is enhancing the online presence of media fragments not only for semantic Web agents, but also traditional search engines. Using Microdata to describe media fragments and publish semantic descriptions as RDF only partially solve this problem. One important reason is that even though Synote, as well as other similar applications, provides synchronised annotations, they are loaded together with the whole multimedia resource on the same physical page. This is reasonable to provide an interactive experience for users. Other than Synote, TED Talks[21] and YouTube interactive transcript allow users to click on the transcript block and the embedded media player on the page will start playing from that time point.

This user-friendly function is not search-engine-friendly for two reasons. Firstly, most search engines only fetch pages as the direct response from the server. So any dynamically generated content on the client-side is ignored. Both Synote Player and YouTube load the transcript by Ajax, but TED generates transcript by server-side script, thus does not have this problem. Secondly, annotations related to media fragments usually share the same replay page with the whole multimedia resource and there is only one URI for the whole page in the search engine index. So it is impossible to link the keywords to a specific media fragment in the search results and users still have to manually find the keyword on the replay page. As the use of schema.org is based on the HTML and search engine indexing infrastructure, we need to make the Web pages, which contain media fragments, search-friendly. One solution is slicing one page into different pages according to media fragments, but the interactive experience will be lost in that users have to open another page to play another media fragment.

Google has developed a framework to crawl Ajax applications[22]. If the token "#!" is included in the original URL, the crawler will know that this page contains Ajax content and in the request URL, "#!" will be replaced by "_es-

[21] `http://www.ted.com/talks`
[22] `http://goo.gl/dPc81`



1: Submit pretty URL "replay/1#!t=3,7" to the crawler
2: Crawler asks the server for "replay/1?_escaped_fragment_=t=3,7"
3: Redirect the request to "snapshot/1?_escaped_fragment_=t=3,7" and the server generates the snapshot page containing annotations and Microdata for "#t=3,7" only. For example, the page only contains keyword "Terrace Threater"
4: The snapshot page is returned by the crawler
5: A user searches "Terrace Threater"
6: Google includes "replay/1#!t=3,7" in the search results
7: The user clicks the link and asks the server for "replay/1#!t=3,7"
8: The server returns the replay page containing all the annotations i.e. both keywords "Terrace Theater" and "linked data"
9: The replay page highlights the media fragment and "Terrace Theater" by playing the fragment from 3s to 7s

Figure 5: The model to improve media fragment presence based on Google crawler

caped_fragment_=" (Google calls it "ugly URL"). On receiving this "Ugly URL" request, the server can return the snapshot page after the dynamic information is fully generated by javascript. The content in snapshot page will be indexed in for the original URL. We design a model on the Synote server to use this framework to index individual snapshot pages for each media fragment. Figure 5 explains how this model could be used to index media fragments. The returned page in step 4 only contains keywords related to fragment "#t=3,7". In the Google index, the original media fragment URLs are associated with the snapshot page. So what Google actually indexed is the URL of the replay page with MFURI syntax attached. Step 8 still returns the whole page, but in step 9, we control the embedded player to play the fragment from 3s to 7s using javascript (see Section 4.1.1) and the corresponding annotations are highlighted straight away. This design not only makes sure media fragments are indexed precisely with the keywords related to it, but also preserves the existing user interface and the interactive experience. This is another reason we choose MFURI syntax as it could be applied to Google's Ajax application crawling framework with slight modification.

5. EVALUATION

This section will present the evaluation results of media fragment publishing in Synote. The evaluation mainly shows the following aspects according to the discussion in Section 4: (1) Media fragments are presented using MFURI syntax and Synote provides both HTML and RDF representation of the media fragments. (2) The published media fragments can be indexed by semantic search engines (we use Sindice and Sig.ma [39] in this evaluation) and interlinked to other datasets. (3) Google can index media fragments and the Microdata can be recognised. Figure 6 shows how we use 303 redirect and content negotiation to dereference HTML and RDF representations of the media fragment as proposed in Section 4.1.1. When asking for application/rdf+xml, the returned RDF file will include the semantic descriptions for all its media fragments. The triples extracted from RDFa in Synmark note resource will also be included in this file as shown in Listing 1. For HTML representation, if the URI is entered directly in the address bar, some browsers, such as Firefox and Google Chrome, will attach the hash fragment after the 303 redirect and the media fragment can be

```
C:\development>curl -iH "Accept:text/html" http://linkeddata
.synote.org/synote/resources/36513#t=1,14
HTTP/1.1 303 See Other
Date: Tue, 14 Feb 2012 18:14:04 GMT
Location: http://linkeddata.synote.org/synote/recording/repl
ay/36513
Content-Type: text/plain; charset=UTF-8
Content-Length: 0
Connection: close

C:\development>curl -iH "Accept:application/rdf+xml" http://
linkeddata.synote.org/synote/resources/36513#t=1,14
HTTP/1.1 303 See Other
Date: Tue, 14 Feb 2012 18:14:10 GMT
Location: /synote/resources/data/36513
Content-Type: text/plain; charset=UTF-8
Content-Length: 0
Connection: close
```

Figure 6: 303 redirect and content negotiation for media fragment dereferencing

highlighted. But in IE and Safari, the fragment is missing.

We have generated a sitemap containing all URIs of resources, media fragments and annotations and submit it to Sindice. Figure 7 shows the semantic description for media fragment *resources/36513#t=00:00:01.000,00:00:14.000* in Sigma. We can see that the triples (lode:illustrate property) embedded in Synmark has been included in the semantic description. In addition, we can check the object URI (the URI of the TED talk page) of lode:illustrate property in Sigma. Figure 8 shows that the lode:involvedAgent and lode:atPlace has been included in the description of the URI from the RDF published in Synote (the "source 1" in Figure 8). So the media fragments published in Synote have been successfully indexed by semantic search engine and interlinked with other datasets through user-generated annotations.

We also submitted the sitemap containing the URIs like *replay/1#!t=3,7* to Google for indexing. To have a quick evaluation of what Googlebot fetches from these URIs, we submit several URIs via Google Web Master Tools[23]. The result shows that on fetching this URL,

`recording/replay/36513#!t=00:00:01.000,00:00:14.000`

the snapshot page is returned with annotations only related to fragment "#t=00:00:01.000,00:00:14.000". The Microdata in the snapshot page can be extracted by Live Microdata[24] and Linter Structured Data[25]. If the keyword

[23] http://www.google.com/webmasters/tools/
[24] http://foolip.org/microdatajs/live/
[25] http://linter.structured-data.org/

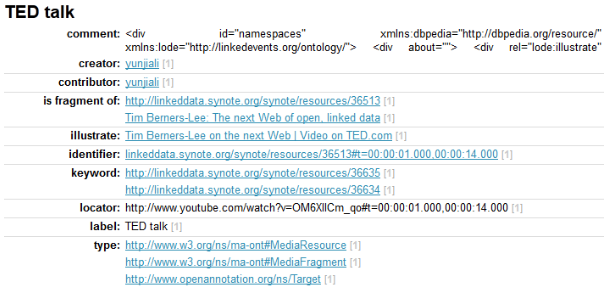


Figure 7: The view of semantic description of media fragments from Sigma

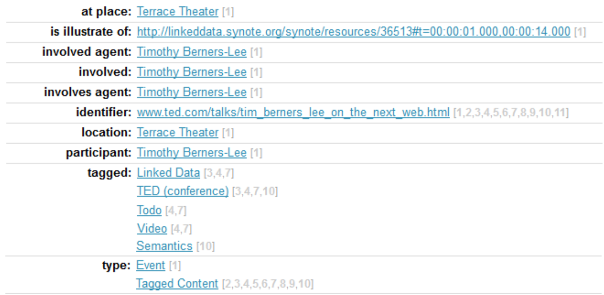


Figure 8: Interlinking with other datasets

"Terrace Theater", for example, is searched in Google, the snapshot page can be successfully found in the search results instead of the whole replay page. When we click the link in search results, the Synote Player page in Figure 2 will be opened and the button in top-left corner indicates that a media fragment is requested. The embedded player will start playing the media fragment "#t=00:00:01.000,00:00:14.000" when the button is clicked and the related annotation on the right column will be highlighted. The screencast[26] and the live demo of this evaluation[27] are available online.

6. CONCLUSION

Media fragments are important for the efficient indexing and searching of multimedia resources. In this paper, we use Synote as the target application to show how media fragments could be published with user-generated annotations following linked data principles. We argue that the semantic descriptions of media fragments can come from two types of sources. Currently, there are plenty of data in Type Two, which is generated by multimedia sharing and annotating applications in Web 2.0 and they should be published as linked data. To publish media fragments and annotations in Synote, we use content negotiation to serve both audio-visual and semantic representation for media fragments. To describe multimedia resources and annotations, we reuse the existing vocabularies online instead of creating new vocabularies. We also suggest to embed RDFa in annotations to promote the interlinking with other datasets.

The design builds an extra layer on top of the existing Synote application. In order to efficiently publish the data for both semantic and traditional search engines, we implement multiple linked data publishing patterns. We also design a model to let Google index media fragments enriched by vocabularies in schema.org. The evaluation shows that requirements listed in Section 3 have been satisfied and the existing Synote still keeps its architecture and user experience. The approach of this work can be further applied to other similar applications, and the solutions could be integrated with multimedia delivery services when the functions proposed in MFURI are implemented on the client and server. In the future, we need to consider other aspects of search engine optimisation to improve the online presence of media fragments. We also need to think about applying this solution to other major search engines such as Yahoo! and Bing. We recommend more media fragments to be published using the solutions described in this paper in order to benefit the indexing of multimedia resources.

[26] `http://goo.gl/4zl1V`
[27] `http://linkeddata.synote.org/synote/`

7. REFERENCES

[1] B. Adida and M. Birbeck. RDFa Primer, Oct. 2008. http://www.w3.org/TR/xhtml-rdfa-primer/.

[2] Adobe Corp. XMP Specification, 2005.

[3] R. Arndt, R. Troncy, S. Staab, L. Hardman, and M. Vacura. COMM: Designing a Well-Founded Multimedia Ontology for the Web. In *International Semantic Web Conference*, volume 4825 of *Lecture Notes in Computer Science*, pages 30–43. Springer Berlin Heidelberg, 2007.

[4] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A Nucleus for a Web of Open Data. In K. Aberer, K.-S. Choi, N. Noy, D. Allemang, K.-I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, editors, *The Semantic Web*, volume 4825 of *LNCS*, pages 722–735. Springer, 2007.

[5] S. Auer, S. Dietzold, J. Lehmann, S. Hellmann, and D. Aumueller. Triplify: light-weight linked data publication from relational databases. *International World Wide Web Conference*, pages 621–630, 2009.

[6] T. Berners-Lee. Linked Data - Design Issues, 2006.

[7] C. Bizer and R. Cyganiak. D2R Server - Publishing Relational Databases on the Semantic Web. In *Proceedings of the 5th International Semantic Web Conference*. Springer, 2006.

[8] J. Bormans and K. Hill. MPEG-21 Overview v.5, 2002.

[9] D. V. Deursen, W. V. Lancker, W. D. Neve, T. Paridaens, E. Mannens, and R. V. D. Walle. NinSuna: a fully integrated platform for format-independent multimedia content adaptation and delivery using Semantic Web technologies. *Multimedia Tools and Applications*, 46(2):371–398, 2009.

[10] O. Erling and I. Mikhailov. RDF Support in the Virtuoso DBMS. *Proceedings of the 1st Conference on Social Semantic Web CSSW*, 221(ISBN 978-3-88579-207-9):59–68, 2007.

[11] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616, June 1999.

[12] B. Haslhofer, W. Jochum, R. King, C. Sadilek, and K. Schellner. The LEMO annotation framework: weaving multimedia annotations with the web.

International Journal on Digital Libraries, 10(1):15–32, 2009.
[13] M. Hausenblas, R. Troncy, T. Bürger, and Y. Raimond. Interlinking Multimedia: How to Apply Linked Data Principles to Multimedia Fragments. *WWW 2009 Workshop Linked Data on the Web LDOW2009*, 2009.
[14] T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*, volume 1. Morgan & Claypool, 2011.
[15] I. Hickson. HTML Microdata, Feb. 2012. http://dev.w3.org/html5/md/.
[16] I. Hickson. WebVTT, Feb. 2012. http://dev.w3.org/html5/webvtt/.
[17] P. Hoschka. An introduction to the synchronized multimedia integration language. *Multimedia, IEEE*, 5(4):84 –88, oct-dec 1998.
[18] I. Jacobs and N. Walsh. Architecture of the World Wide Web. *arq Architectural Research Quarterly*, 2(01):1–56, 2004.
[19] J. Kahan, M. Koivunen, E. Prud'Hommeaux, and R. Swick. Annotea: An open rdf infraestructure for shared web annotations. In *Proceedings of the 10th international conference on World Wide Web*, pages 623–632. ACM Press, 2001.
[20] D. Lambert and H. Q. Yu. Linked Data Based Video Annotation and Browsing for Distance Learning. In *The 2nd International Workshop on Semantic Web Applications in Higher Education (SemHE'10)*, Nov. 2010.
[21] Y. Li, M. Wald, and G. Wills. Interlinking multimedia annotations. In *ACM WebSci'11*, pages 1–4, June 2011. WebSci Conference 2011.
[22] Y. Li, M. Wald, G. Wills, S. Khoja, D. Millard, J. Kajaba, P. Singh, and L. Gilbert. Synote: development of a web-based tool for synchronized annotations. *New Review of Hypermedia and Multimedia*, 17(3):295–312, 2011.
[23] E. Mannens, D. Van Deursen, R. Troncy, S. Pfeiffer, C. Parker, Y. Lafon, J. Jansen, M. Hausenblas, and R. Van de Walle. A uri-based approach for addressing fragments of media resources on the web. *Multimedia Tools and Applications*, pages 1–25. 10.1007/s11042-010-0683-z.
[24] F. Manola and E. Miller. RDF Primer, 2004.
[25] J. M. Martinez. MPEG-7 Overview, 2004.
[26] S. Pfeiffer, C. D. Parker, and A. T. Pang. Specifying Time Intervals in URI Queries and Fragments of Time-Based Web Resources, 2005.
[27] E. Prud'hommeaux and A. Seaborne. SPARQL Query Language for RDF, 2008.
[28] C. Saathoff and A. Scherp. Unlocking the semantics of multimedia presentations in the web with the multimedia metadata ontology. *Proceedings of the 19th international conference on World wide web WWW 10*, page 831, 2010.
[29] F. Sasaki, T. Bürger, J. Söderberg, V. Malaisé, F. Stegmaier, and W. Lee. Ontology for media resource 1.0, Feb. 2012. http://www.w3.org/TR/mediaont-10/.
[30] L. Sauermann and R. Cyganiak. Cool URIs for the Semantic Web. *W3C Note*, 49(681):1–15, 2008.
[31] J. F. Sequeda and O. Corcho. Linked Stream Data: A Position Paper. In K. Taylor, A. Ayyagari, and D. D. Roure, editors, *International Workshop on Semantic Sensor Networks 2009*, volume 522, pages 148–157. CEUR, Kerry Taylor and Arun Ayyagari and David De Roure, 2009.
[32] R. Shaw, R. Troncy, and L. Hardman. Lode: Linking open descriptions of events. *The Semantic Web*, 5926(Lecture Notes in Computer Science):153–167, 2009.
[33] T. Steiner. SemWebVid - Making Video a First Class Semantic Web Citizen and a First Class Web Bourgeois. In *Proceedings of the 9th International Semantic Web Conferene ISWC 2010*, pages 3–6. Springer, 2010.
[34] T. Steiner, R. Troncy, and M. Hausenblas. How Google is using Linked Data Today and Vision For Tomorrow. In S. Auer, S. Decker, and M. Hauswirth, editors, *Linked Data in the Future Internet 2010*, Ghent,Belgium, 2010.
[35] Technical Standardization Committee on AV & IT Storage Systems and Equipment. Exchangeable Image File Format for Digital Still Cameras: EXIF Version 2.2, 2002.
[36] R. Troncy. Bringing the IPTC News Architecture into the Semantic Web. *Lecture Notes In Computer Science Vol 5318*, 5318:483, 2008.
[37] R. Troncy, L. Hardman, and J. V. Ossenbruggen. Identifying spatial and temporal media fragments on the web. *W3C Video on the Web Workshop*, pages 4–9, 2007.
[38] R. Troncy, E. Mannens, S. Pfeiffer, and D. V. Deursen. Media fragments URI 1.0 (basic), Mar. 2012. http://www.w3.org/TR/media-frags/.
[39] G. Tummarello, R. Cyganiak, M. Catasta, S. Danielczyk, R. Delbru, and S. Decker. Sig.ma: Live views on the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(4):355 – 364, 2010.
[40] G. Tummarello, R. Delbru, and E. Oren. Sindice.com: Weaving the open linked data. In K. Aberer, K.-S. Choi, N. Noy, D. Allemang, K.-I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, editors, *The Semantic Web*, volume 4825 of *Lecture Notes in Computer Science*, pages 552–565. Springer Berlin / Heidelberg, 2007. 10.1007/978-3-540-76298-0_40.
[41] M. Verwaest, D. Van Deursen, R. De Sutter, N. Laukens, D. Van Rijsselbergen, E. Mannens, and R. Van de Walle. Television on a tablet : how tv will look like in the future and how to deal with professionally produced content. In *Second W3C web and TV workshop : papers*, page 8. W3C, 2011.
[42] J. Waitelonis. Augmenting video search with linked open data. In *Proceedings of International Conference on Semantic Systems (i-semantics), September 2-4, 2009, Graz, Austria*. Verlag der TU Graz, Austria, 2009.