



NoTube

*Networks and ontologies for the transformation and unification of broadcasting
and the Internet*

FP7 – 231761

D5.4 Design of eCommerce-Oriented Video Annotation Environment

Coordinator: Dong Liu (OU)

With contributions from: H. Qing Yu (OU)

Quality Assessor: Libby Miller

Quality Controller: John Domingue

| | |
|----------------------|-----------------------|
| Document Identifier: | NoTube/2011/D5.4/v1.2 |
| Version: | 1.2 |
| Date: | 31/10/2011 |
| State: | Final |
| Distribution: | PU |

EXECUTIVE SUMMARY

This deliverable reports the requirement analysis and the design of “Watch ‘n’ Buy”, an eCommerce-oriented video annotation environment. While enabling annotating video clips with products, “Watch ‘n’ Buy” aims at realizing a novel advertisement insertion model that allows customers purchasing products they see on the screen. Moreover, it integrates advertisements and video clips in a new manner. Audience will not be forced to watch any advertisements along with programmes. Instead, when they are interested in the products appearing on the screen and pause the video, the annotations will be displayed and indicate links to retailers of the products. In other words, the playback of video clips and advertisements are in parallel rather than sequentially. In addition, “Watch ‘n’ Buy” also provides social networking functionalities such as followers and feeds, so that the annotations can be rapidly propagated.

From the technical perspective, a major distinctive feature of “Watch ‘n’ Buy” is the fact that it follows the Linked Services approach, not only for facilitating the harvesting of metadata of both videos and products, but also for ensuring the flexibility, extendibility and robustness. Most of the components of “Watch ‘n’ Buy” such as semantic extraction services, will be implemented as Web services, and whose external interfaces will be semantically described and published as Linked Data. On the other hand, with the help of the Linked Services infrastructure, “Watch ‘n’ Buy” is able to select and invoke appropriate services on the fly. In order to cater for interoperability, we re-use widespread ontologies and vocabularies in the domains of eCommerce, social Web and multimedia to construct the conceptual model of video annotation. Finally, for being a typical semantic Web application, “Watch ‘n’ Buy” publishes video annotations as Linked Data on the Web and exposes a standards compliant SPARQL endpoint.

DOCUMENT INFORMATION

| | | | |
|---------------------------|---|----------------|--------|
| IST Project Number | FP7 - 231761 | Acronym | NoTube |
| Full Title | Networks and ontologies for the transformation and unification of broadcasting and the Internet | | |
| Project URL | http://www.notube.tv | | |
| Document URL | | | |
| EU Project Officer | Leonhard Maqua | | |

| | | | | |
|---------------------|---------------|-----|--------------|---|
| Deliverable | Number | 5.4 | Title | Design of eCommerce-Oriented Video Annotation Environment |
| Work Package | Number | 5 | Title | Semantic TV Resource Broker |

| | | | | |
|----------------------------|--|-----|---------------|-------------------------------------|
| Date of Delivery | Contractual | M33 | Actual | |
| Status | version 1.2 | | final | <input checked="" type="checkbox"/> |
| Nature | prototype <input type="checkbox"/> report <input checked="" type="checkbox"/> dissemination <input type="checkbox"/> | | | |
| Dissemination level | public <input checked="" type="checkbox"/> consortium <input type="checkbox"/> | | | |

| | | | | |
|---------------------------|----------------|----------|---------------|--|
| Authors (Partner) | OU | | | |
| Responsible Author | Name | Dong Liu | E-mail | d.liu@open.ac.uk |
| | Partner | OU | Phone | +44 (0)1908-858216 |

| | |
|-------------------------------------|---|
| Abstract (for dissemination) | <p>This deliverable reports the requirement analysis and the design of “Watch ‘n’ Buy”, an eCommerce-oriented video annotation environment. While enabling annotating video clips with products, “Watch ‘n’ Buy” aims at realizing a novel advertisement insertion model that allows customers purchasing products they see on the screen. Moreover, it integrates advertisements and video clips in a new manner. Audience will not be forced to watch any advertisements along with programmes. Instead, when they are interested in the products appearing on the screen and pause the video, the annotations will be displayed and indicate links to retailers of the products. In other words, the playback of video clips and advertisements are in parallel rather than sequentially. In addition, “Watch ‘n’ Buy” also provides social networking functionalities such as followers and feeds, so that the annotations can be rapidly propagated.</p> |
| Keywords | Video Annotation, eCommerce, Linked Services, Social Networking, Advertisement Insertion |

| Version Log | | | |
|--------------------|-----------------|--------------------|---------------------------------|
| Issue Date | Rev. No. | Author | Change |
| 16/09/2011 | 0.1 | D. Liu | Preliminary ToC |
| 07/10/2011 | 0.2 | D. Liu | Section 1, 2 |
| 19/10/2011 | 0.3 | D. Liu, H. Qing Yu | Section 3 |
| 24/10/2011 | 0.4 | D. Liu | Section 4 |
| 25/10/2011 | 0.6 | D. Liu | Section 5 |
| 26/10/2011 | 0.8 | D. Liu, H. Qing Yu | Section 6, 7 |
| 27/10/2011 | 1.0 | D. Liu | Overall refinement |
| 31/10/2011 | 1.2 | D. Liu | Overall refinement and Appendix |

PROJECT CONSORTIUM INFORMATION

| Participants | | Contact |
|---|---|--|
| Vrije Universiteit Amsterdam |  | Guus Schreiber Phone: +31 20 598 7739/7718 Email: schreiber@cs.vu.nl |
| British Broadcasting Corporation |  | Libby Miller Phone: +44 787 65 65 561 Email: Libby.Miller@bbc.co.uk |
| Pro-netics |  | Marco Bruni Phone: +39 06 45472503 Email: marco.bruni@pro-netics.com |
| Engin Medya Hizmetleri A.S. |  | Ron van der Heiden Phone: +31 6 2003 2006 Email: ron@engin.tv |
| Institut fuer Rundfunktechnik GmbH |  | Christoph Dosch Phone: +49 89 32399 349 Email: dosch@irt.de |
| Ontotext AD |  | Atanas Kiryakov Phone: +35 928 091 565 Email: naso@sirma.bg |
| Open University |  | Stefan Dietze Phone: +44 1908 858 217 Email: s.dietze@open.ac.uk |
| RAI Radiotelevisione Italiana SPA |  | Alberto Morello Phone: +39 011 810 31 07 Email: a.morello@rai.it |
| Semantic Technology Institute International |  | Lyndon Nixon Phone: +43 1 23 64 002 Email: lyndon.nixon@sti2.org |
| Stoneroos B.V. |  | Annelies Kaptein Phone: +31 35 628 47 22 Email: annelies.kaptein@stoneroos |
| Thomson Video Networks |  | Raoul Monnier Phone: +33 2 99 27 30 57 Email: raoul.monnier@thomson-networks.com |
| Polymedia, SpA |  | Tullio Pirovano Phone: +39 02 257711 Email: tullio.pirovano@polymedia.it |
| KT Corporation |  | Myoung-Wan Koo Phone: +82 2 526 6347 Email: mskim@kt.co.kr |

TABLE OF CONTENTS

| | |
|---|-----------|
| LIST OF FIGURES | 8 |
| LIST OF ACRONYMS..... | 9 |
| 1. INTRODUCTION..... | 10 |
| 1.1. MOTIVATION..... | 10 |
| 1.2. VISION..... | 10 |
| 2. REQUIREMENTS..... | 12 |
| 2.1. VIDEO ANNOTATION..... | 12 |
| 2.2. METADATA HARVESTING | 13 |
| 2.3. SEMANTICS EXTRACTION AND METADATA ENRICHMENT | 13 |
| 2.4. SOCIAL NETWORKING FUNCTIONALITIES..... | 13 |
| 2.5. NON-FUNCTIONAL REQUIREMENTS..... | 14 |
| 2.6. INTEGRATION WITH NOTUBE USE CASES..... | 14 |
| 3. CONCEPTUAL MODEL..... | 15 |
| 3.1. ANNOTATION | 15 |
| 3.2. VIDEO | 16 |
| 3.2.1. W3C MEDIA ONTOLOGY | 16 |
| 3.2.2. BBC PROGRAMMES ONTOLOGY..... | 16 |
| 3.2.1. TIME AND TIMELINE ONTOLOGY..... | 16 |
| 3.3. PRODUCT..... | 16 |
| 3.3.1. GOODRELATIONS ONTOLOGY | 16 |
| 3.3.2. TAXONOMY..... | 17 |
| 3.4. SOCIAL NETWORKING..... | 17 |
| 3.5. PROVENANCE..... | 17 |
| 4. OVERALL ARCHITECTURE | 18 |
| 4.1. ARCHITECTURE OVERVIEW | 18 |
| 4.2. COMPONENTS..... | 18 |
| 4.2.1. RDF DATA REPOSITORY..... | 19 |
| 4.2.2. VIDEO METADATA IMPORTER | 19 |
| 4.2.3. PRODUCT METADATA IMPORTER | 19 |
| 4.2.4. WATCH 'N' BUY PLAYER..... | 19 |
| 4.2.5. WATCH 'N' BUY PORTAL..... | 20 |
| 4.2.6. LINKED DATA PROVIDER | 20 |
| 4.2.7. OTHER COMPONENTS | 20 |
| 5. EXTERNAL INTERFACES..... | 21 |
| 5.1. WATCH 'N' BUY PLAYER..... | 21 |
| 5.1.1. WEB BASED USER INTERFACE | 21 |
| 5.1.2. USER INTERFACE ON THE SECOND SCREEN..... | 21 |
| 5.2. WATCH 'N' BUY PORTAL..... | 22 |
| 5.3. RESTFUL API..... | 22 |
| 5.3.1. USER | 23 |
| 5.3.2. ANNOTATION | 23 |
| 5.3.3. PRODUCT..... | 24 |
| 5.3.4. VIDEO | 25 |
| 5.3.5. COMMENT | 25 |
| 5.3.6. RATING | 26 |
| 5.3.7. TAG..... | 27 |
| 5.4. SPARQL ENDPOINT | 27 |

| | |
|---|-----------|
| 6. RELATED WORK | 28 |
| 6.1. ANNOMATION | 28 |
| 6.2. SUGARTUBE..... | 29 |
| 6.3. THINGS YOU SAW IN A MOVIE | 29 |
| 6.4. TESCO'S BILLBOARD GROCERY STORE..... | 29 |
| 7. CONCLUSION | 31 |
| 8. REFERENCES..... | 32 |
| A. APPENDIX A | 33 |

List of Figures

| | |
|--|----|
| Figure 1. Conceptual model of video annotation | 15 |
| Figure 2. The conceptual sketch of the “Watch ‘n’ Buy” | 18 |
| Figure 3. Software components of “Watch ‘n’ Buy” | 19 |
| Figure 4. Mockup of Web-based “Watch ‘n’ Buy” player..... | 21 |
| Figure 5. Mockup of “Watch ‘n’ Buy” player on the second screen..... | 21 |
| Figure 6. Mockup of “Watch ‘n’ Buy” portal | 22 |
| Figure 7. Screenshot of Annomation..... | 28 |
| Figure 8. Screenshot of SugarTube | 29 |

List of Acronyms

| Acronym | Description |
|----------------|--|
| API | Application Programming Interface |
| ASIN | Amazon Standard Identification Number |
| CRUD | Create, Read, Update and Delete |
| FOAF | Friend-of-a-Friend |
| GUI | Graphic User Interface |
| HTML | HyperText Markup Language |
| HTTP | HyperText Transfer Protocol |
| LOD | Linked Open Data |
| NAISC | North American Industry Classification System |
| OWL | Web Ontology Language |
| QR | Quick Response |
| RDF | Resource Description Framework |
| RDFS | RDF Schema |
| SIOC | Semantically-Interlinked Online Communities |
| SMS | Short Message Service |
| SWS | Semantic Web Services |
| UNSPC | United Nations Standard Products and Services Code |
| URL | Uniform Resource Locator |
| W3C | World Wide Web Consortium |
| WP | Working Package |
| WYSIWCB | What You See Is What You Can Buy |
| XML | eXtensible Markup Language |
| XSLT | Extensible Stylesheet Language Transformations |

1. Introduction

1.1. Motivation

Advertisements are now ubiquitous, appearing at corners of the screen or being placed directly in programmes (a.k.a. embedded marketing). However, most of them are not interactive. For instance, when a person is interested in some soft drink on TV (in an advertisement or product placement), how can s/he buy one? S/he needs to search for a shop selling the soft drink, which might be difficult and time consuming. Also it is not desirable that a customer or consumer should have to take responsibility for finding products that s/he wants to buy.

On the other hand, advertisements attached to TV programmes or online videos are disturbing to a certain extent. One of the reasons is that those advertisements are always visible to all the audience, and are played sequentially, before or after TV programmes. We can improve the user experience if the advertisements are “played” in parallel with programmes, and only become visible at a user request.

In addition, explicit or implicit celebrity endorsement is a very efficient advertising channel. For example, Kate Middleton’s Royal Wedding Ring is now a very popular gift in the UK. If both consumers and manufactures are able to annotate TV programmes and video clips, it will be easy for them to establish connections with each other.

This application, “Watch ‘n’ Buy”, aims to address these issues. The idea is to develop a platform that allows both vendors and consumers identifying and annotating products appear on the screen, which they can offer or they are interested in. Different from the conventional way of product placement, “Watch ‘n’ Buy” is a social networking application open to the public. The annotations of products are added after the production of a programme, and will be displayed only when the programme is pause. Furthermore, our ultimate goal is to implement WYSIWCB: What You See (on TV, in online video, in poster, etc.) Is What You Can Buy. We will first focus on TV programmes and online videos that are formally described by some metadata.

1.2. Vision

Our vision on eCommerce-oriented video annotation is to implement a novel model of advertisement insertion. Generally, there have been two existing ways of integrating advertisements and videos. One is playing advertisements at the beginning or the end of programmes, or showing them at a corner of the screen. The other way is product placement, which, from the audience’s viewpoint, is less disturbing than the former. However, product placement cannot provide the crucial information needed to purchase the products. Even more, sometimes only the logos rather than actual products appear in videos or programmes, which makes it much more difficult for customers to search for products. The novel way of advertisement insertion we attempt to implement is to attach semantic descriptions of products to the programmes. If a user is interested in a product on the screen, s/he just need to pause the programme and follows a link to the vendor to buy it. The ideal user experience would be like grabbing things directly from the screen. Both the audience and advertisers will benefit from the proposed advertisement insertion model. For audience, programmes will be devoid of advertisements. For advertisers, there will be more spaces for placing advertisements.

The scenarios we considered are as follows:

Product manufacturers and retailers can employ advertising agents to annotate video clips and programmes with descriptions of their products. Moreover, they also can make the annotations by utilizing crowd power, if they give, for example, some cash back or vouchers to those who annotate videos with their products. In order words, “Watch ‘n’ Buy” facilitates the crowdsourcing of image recognition and video annotation.

Video makers can earn incomes by annotating their video clips with some products. Actually, video-hosting websites like YouTube have started to share incomes from advertisements with artists who made the videos¹. It would be also a possible way for artists to rise funding for a video: vendors may want to be the sponsors, if the artists annotating the videos with their products

Fashion designers can build the reputation of their brands. Suppose that a designer, who is not very well known, accidentally realises that a celebrity in a TV programme is wearing his (or her) work. The designer would be able to significantly increase the sale, if s/he can annotate the programme with the link to an online store selling the cloth.

¹ <http://www.youtube.com/partners>

2. Requirements

This section enumerates functional and non-functional requirements for “Watch ‘n’ Buy”. As mentioned before, we envision “Watch ‘n’ Buy” to be a social semantic Web application that enables collaboratively annotating video clips and TV programmes with descriptions of products. “Watch ‘n’ Buy” can extract metadata and implicit semantics of both videos and products, and also can publish them in a machine-understandable way on the Web.

2.1. Video Annotation

Video annotation is the most important functionality of “Watch ‘n’ Buy”, which essentially allows annotating video clips with the products that a user is selling, saw or is seeking for. Similar to sharing links on Facebook, what users need to do for annotating a video with a product they sell or saw are: 1) pause the video; 2) select a region on the screen; and 3) enter the URL of the product. Using a Web browser with support for HTML 5², the users can directly drag and drop the URL onto the screen. Before saving and publishing the new-added annotation, “Watch ‘n’ Buy” first validates if the provided URL points to the description of a product. If not, the annotation will be rejected so as to prevent the system from spam.

If a user find an interesting product in a video, yet has not been annotated yet, the user can add a specific annotation to it, which indicates that s/he is seeking for it. “Watch ‘n’ Buy” will carry out the following steps to help the user in find the product:

- Search the local repository of product information according to the user input
- Search external repositories by invoking Web services provided by electronic commerce companies such as eBay and Amazon
- Send requests to relevant retailers

If the system is able to find out products matching the user’s needs, it will automatically send notification to the user via email, SMS or other methods. If not, the “Watch ‘n’ Buy” system will publish the user request as Linked Data on the Web. Once a vendor replies to the request, the user will get a notification.

The “Watch ‘n’ Buy” application should track the role that an annotator plays, e.g. as a seller, a buyer or third party. And, it also should distinguish products and the offers of them, which means a certain product can be connected to various offers provided by different vendors. However, a product can only have one single set of metadata that comprehensively describes it. The objective of this mechanism is to avoid too many annotations to be displayed on the same screen.

In addition, “Watch ‘n’ Buy” should have the capability of capturing the spatio-temporal properties of annotations. Annotations not only can occupy an arbitrary region of the screen, but can span a period of the timeline of the video as well. “Watch ‘n’ Buy” should offer the flexibility of representing video annotations at proper position and at correct time, when the video being played on different devices with screens of different resolutions. Therefore, it is critical to keep the spatio-temporal information in a unified format.

To leverage the self-correcting nature of crowdsourcing to improve the accuracy of annotation, “Watch ‘n’ Buy” should allow the users modifying the position (either on screen or on the timeline) and the metadata of an annotation, as well as adding comments to it. The collaborative image recognition can be conducted in this way, and, gradually, the annotations will more and more precisely depict the products emerging in a video clip.

² <http://www.w3.org/TR/html5/>

2.2. Metadata Harvesting

In this deliverable, metadata harvesting refers to the collection of descriptive information about video clips, TV programmes and products from distributed sources through the invocation of corresponding Linked Services.

“Watch ‘n’ Buy” should realise both automated and semi-automated mechanisms of metadata harvesting. For the websites, e.g. YouTube³, BBC⁴, etc., that offer dedicated libraries, services or APIs to facilitate metadata harvesting, “Watch ‘n’ Buy” should take advantage of them to retrieve the metadata as much as possible. On the other hand, if the data sources exist in the format of HTML, “Watch ‘n’ Buy” should be able to make use of transformation mechanisms such as XSLT to scrape metadata from HTML pages.

2.3. Semantics Extraction and Metadata Enrichment

The objective of semantics extraction is to formalise the results of metadata harvesting in a machine-understandable way. It enables automatic classification of products, and also lays the foundation for the aforementioned validation of annotation. The formalisation of metadata should be aligned with domain-specific ontologies. For instance, BBC Programmes Ontology⁵, W3C Media Ontology⁶ and GoodRelations⁷ are respectively suitable for representing the explicit semantics of TV programmes, video clips and products.

Another task needs to be performed is the interlinking with other resources on the Web of Data. For example, products identified by ASIN (Amazon Standard Identification Number) can be linked to the corresponding ones stored in productdb⁸. In addition, semantics can also be imported from general-purpose Linked Data ingestion services, such as URIBurner⁹, which take URLs as input and produce Linked Data Objects graphs. However, the Linked Data gotten from URIBurner may be of bad quality¹⁰, i.e. contain irrelevant data, or have a long response time. There are the issues need to be addressed by adopting some validation and caching mechanisms.

As for user requests, i.e. the products they are seeking for, text-based metadata enrichment services such as Zamanta¹¹, Lupedia¹² and Sindice¹³, need to be executed to help machines in understanding the user requests. For example, when a user annotates a video clip about the Royal Wedding with a word “ring”, the “Watch ‘n’ Buy” system should be able to know that word implies the wedding ring rather than a circle.

2.4. Social Networking Functionalities

As a social Web application, “Watch ‘n’ Buy” should implements the following social networking functionalities:

- **Follow.** This functionality includes both asymmetric and symmetric following. In other words, a user can unilaterally subscribe to the annotations posted by any number of other individuals and groups.
- **Feed.** This functionality provides users with updated information about their annotations. For example, it notifies a user that there is a retailer selling the product s/he is seeking for.

³ http://code.google.com/apis/youtube/2.0/developers_guide_protocol_audience.html

⁴ <http://www0.rdthdo.bbc.co.uk/services/>

⁵ <http://www.bbc.co.uk/ontologies/programmes/2009-09-07.shtml>

⁶ <http://www.w3.org/TR/mediaont-10/>

⁷ <http://purl.org/goodrelations/v1>

⁸ <http://productdb.org/>

⁹ <http://uriburner.com/>

¹⁰ See this as an example for bad quality Linked Data gotten from URIBurner:

<http://linkeddata.uriburner.com/ode/?uri=http://www.amazon.com/gp/product/0553383043>

¹¹ <http://developer.zemanta.com/docs/>

¹² <http://lupedia.ontotext.com/>

¹³ <http://sindice.com/developers/api>

- **Review.** Reviewing includes rating, tag and comments can be made on annotations, products and video clips.
- **Sharing.** This refers to sharing annotations onto other social networking websites like Twitter, Facebook. It should be attached with a screenshot of the related video clip.

2.5. Non-Functional Requirements

Several non-functional requirements are collected, which are necessary for providing the eCommerce-oriented video annotation environment.

- **Usability.** “Watch ‘n’ Buy” needs to be usable by common end-users. This is important to provide the possibility of everyone to annotate video clips in an easy way.
- **Flexibility.** Users should be able to access the system using various devices, e.g. Web browsers, smartphones, tablet computers, etc.
- **Extendibility.** “Watch ‘n’ Buy”, especially the metadata harvesting functionality, should be extendable in order to import data from more sources.
- **Robustness.** When an external service is temporarily unavailable, “Watch ‘n’ Buy” should be able to smoothly switch an alternative. For instance, if Sindice is unavailable, the system should be aware of it and use Lupedia instead.
- **Error handling.** Whenever an error occurs, users should receive a detailed and human readable error message with the explanation of the reasons for the error and possible solutions to it.

2.6. Integration with NoTube Use Cases

Outcomes from other working packages of the NoTube project can be integrated with “Watch ‘n’ Buy”. For example, it will significantly enhance the flexibility of the “Watch ‘n’ Buy” system, if being integrated with the TEA (TV Extras Authoring) Player. In that way, the functionalities of “Watch ‘n’ Buy” will also be accessible for the users of the N screen scenario. In addition, video content analysis components developed in WP4 can help in the improvement of the user experiences by automating the detection of the regions of interests.

3. Conceptual Model

This section introduces the conceptual model of “Watch ‘n’ Buy”, the design of which is guided by two principles: 1) reusing as much as possible the widespread domain-specific ontologies and vocabularies; 2) balancing between expressiveness and complexity. According to requirements presented in Section 2, the proposed conceptual model covers five aspects: annotation, video, product, social networking and provenance, which are respectively elaborated in the rest of this section.

3.1. Annotation

The modelling of video annotation is on the basis of our previous work of Annomation¹⁴. Annomation is a semantic video annotation platform for annotating and publishing educational video resources using Linked Data identifiers and free text. For the purpose of its application in the domain of eCommerce, we extend and refine the video annotation ontology of Annomation.

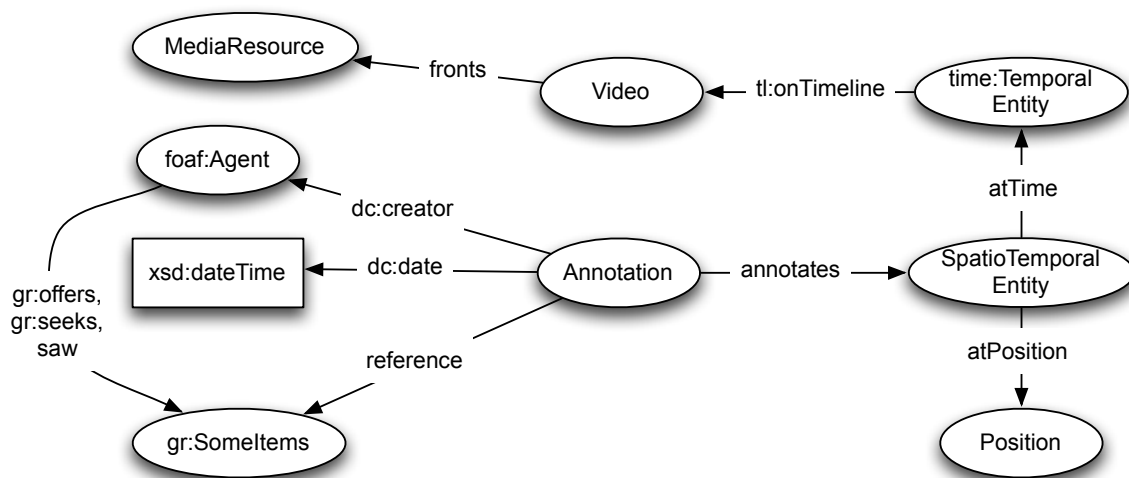


Figure 1. Conceptual model of video annotation

As shown in Figure 1, the core of video annotation ontology is the concept *Annotation*. Different from the ontology of Annomation, this ontology introduces a new concept called *SpatioTemporalEntity* to capture the positions of an annotation both on the screen and on timeline of a video clip. In more detail, with the help of *time:TemporalEntity*, which is adopted from W3C Time Ontology¹⁵, annotations can be created for a instant time or time duration on a video. The concept *Position* is defined for capturing the location of an annotation on the screen. And, it is open to coordinate systems, which can be 2-deminsional, 3-deminsional or regularization coordinates. Similarly, the concept *Video* serves as the common ancestor of different kinds of video streams including online video clips, TV programmes, movies, etc.

The concepts, *foaf:Agent* and *gr:SomeItems*, are respectively borrowed from two widely used vocabularies: Friend-of-a-Friend (FOAF)¹⁶ and GoodRelations¹⁷. Originally, GoodRelations provides only two properties to link business entities and offerings, i.e. *gr:offers* and *gr:seeks*. The former is connected to the sale side, and the latter is to the buy side. We introduce another property *saw* to link a product to a third-party annotator.

¹⁴ <http://annomation.open.ac.uk/annomation>

¹⁵ <http://www.w3.org/TR/owl-time/>

¹⁶ <http://xmlns.com/foaf/spec/>

¹⁷ <http://purl.org/goodrelations/v1>

3.2. Video

In a broad context, a video is a sequence of images in motion, which includes movies, TV programmes and online video clips. Although “Watch ‘n’ Buy” currently focuses on the annotations on TV programmes and online videos, a generic media ontology, i.e. W3C Media Ontology¹⁸, is adopted to ensure the applicability and extendibility. In addition, BBC Programmes Ontology is used for modelling the metadata of TV programmes. Two auxiliary ontologies, Time and Timeline, are used for identifying temporal instants and durations on the video timeline.

3.2.1. W3C Media Ontology

As W3C Media Ontology aims to bridge among heterogeneous models of multimedia resource, it provides the core vocabulary mainly for describing video and audio tracks. Some of the concepts and properties we borrowed from W3C Media Ontology are enumerated as follows: *MediaResource*, *title*, *identifier*, *duration*, *description*, *frameHeight*, *frameWidth*, etc. More details about this ontology can be found in the specification¹⁸.

A benefit brought by W3C Media Ontology is that a number of schemas mappings to the existing metadata formats have already been established (see Section 5.2 of the specification of W3C Media Ontology). For example, the YouTube Data API protocol¹⁹ can be completely mapped to W3C Media Ontology.

3.2.2. BBC Programmes Ontology

BBC Programmes Ontology²⁰ aims to model programmes with a set of concepts and properties covering different aspects of programmes, e.g. content, medium, publishing, categorisation, etc. This ontology provides domain-specific terms like *Brand*, *Series*, *Episode*, which are useful for capturing the metadata of TV programmes.

Additionally, BBC Programmes Ontology is built by re-using ontologies such as FOAF and Timeline Ontology. Therefore, it shares a common ground with other modules of the proposed conceptual model, and can be easily aligned with those modules, especially the annotation module presented in Section 3.1.

3.2.1. Time and Timeline Ontology

As mentioned before, W3C Time Ontology²¹ and the Timeline Ontology²² are two auxiliary components for modelling the temporal properties of video annotations. We use *time:Instant* and *time:Interval* to respectively represent a time point and a period of time, and we exploit the Timeline Ontology to formally describe the stream of a video or a TV programme.

3.3. Product

For GoodRelations²³ is a widely used ontology in the area of eCommerce, it is employed to model the metadata of products and their offers. Another important issue of describing products is the categorization. To this end, we allow users to assign a product to a category of arbitrary taxonomy.

3.3.1. GoodRelations Ontology

The vocabulary provided by GoodRelations is enough for most of the use cases in the area of eCommerce, for it covers most of the descriptive and commercial aspects of describing products and services [1], e.g. pricing, payment, location, shipment, delivery, etc. It is worth noting that the GoodRelations Ontology can be interpreted by a reasoner on the level of RDFS, which reduces the overhead of reasoning and querying the semantic descriptions of products.

¹⁸ <http://www.w3.org/TR/mediaont-10/>

¹⁹ <http://code.google.com/intl/en/apis/youtube/2.0/reference.html>

²⁰ <http://www.bbc.co.uk/ontologies/programmes/2009-09-07.shtml>

²¹ <http://www.w3.org/TR/owl-time>

²² <http://purl.org/NET/c4dm/timeline.owl>

²³ <http://purl.org/goodrelations/v1>

The challenge of using GoodRelations as the conceptual model of products is the lack of importers that can transform the HTML pages of different sources into RDF statements. Good news is that a micro-format named hProduct²⁴ has been proposed and applied in several websites²⁵. However, several large eCommerce websites such as Amazon, eBay have not added hProduct annotations to their HTML pages of product description. Thus, an open platform is required to allow developers sharing the metadata ingestion services.

3.3.2. Taxonomy

As stated before, “Watch ‘n’ Buy” is open to arbitrary taxonomies of products and services, e.g. eClass²⁶, NAICS (North American Industry Classification System)²⁷, UNSPC (United Nations Standard Products and Services Code)²⁸. To assign a product to a certain category, the annotator need to link the product to the URL of the category with the predicate of <http://www.w3.org/2006/vcard/ns#category>.

3.4. Social Networking

FOAF vocabulary provides the means of user identification and profiling on the semantic level. Hence it is used to model the user information. Since the annotations on video clips are regarded as a special kind of micro-blog posts, SIOC²⁹ is adopted to capture the relevant metadata. In this case, the concept *Annotation* is defined as a sub-class of *sioc:MicroblogPost*. In addition, CommonTag³⁰ and the reviewing terms defined by Revyu³¹ are respectively used to model the tags and reviews.

3.5. Provenance

The adoption of the Provenance Vocabulary³² aims to describe the fact that who, when and how a user created an annotation. Besides that, the provenance information about metadata harvesting is also going to be collected, namely the system will track when and by which means the metadata about the involved video clips and products is cropped.

²⁴ <http://microformats.org/wiki/hproduct>

²⁵ <http://microformats.org/wiki/hproduct-examples>

²⁶ <http://www.heppnetz.de/projects/eclassowl/>

²⁷ <http://www.census.gov/eos/www/naics/>

²⁸ <http://www.daml.org/2004/05/unspsc/unspsc.owl>

²⁹ <http://rdfs.org/sioc/spec/>

³⁰ <http://commontag.org/Home>

³¹ <http://vocab.org/review/terms.html>

³² <http://trdf.sourceforge.net/provenance/ns.html>

4. Overall Architecture

This section overviews the architecture of “Watch ‘n’ Buy”, which is designed following the paradigm of Linked Services (see D5.2 and D5.3). In short, the underlying components, e.g. metadata importers and validators, are wrapped as Semantic Web Services (SWS), whose descriptions are published as Linked Data. With the support of Linked Services infrastructure, “Watch ‘n’ Buy” can choose proper services and invoke them.

4.1. Architecture Overview

Figure 2 shows the conceptual sketch of the “Watch ‘n’ Buy” application. Sitting in the centre, iServe 错误! 未找到引用源。 represents the whole Linked Services infrastructure (more information can be found in D5.2). Dedicated services will be invoked through the infrastructure to gather the metadata from eCommerce websites such as eBay, Amazon, as well as from video sharing websites like YouTube. Our previous work has wrapped the metadata retrieval and enrichment services developed in NoTube project as Linked Services (see D5.2), thus they can be integrated to the “Watch ‘n’ Buy” system. A key feature of Linked Services is that their outputs are in the format of RDF, which is suitable for feeding semantic Web applications like “Watch ‘n’ Buy”.



Figure 2. The conceptual sketch of the “Watch ‘n’ Buy”

4.2. Components

In this section, we identify the architectural components of “Watch ‘n’ Buy”. Figure 3 depicts a logical view of the overall architecture with three layers:

- L1: Infrastructure Layer
- L2: Core Layer
- L3: Interface Layer

The Infrastructure Layer (L1) provides supportive functionalities such as data repository and the discovery and execution of Linked Services. The implementation of L1 mainly relies on three components: a RDF repository, iServe and OmniVoke 错误! 未找到引用源。 . The Core Layer (L2) consists of the components that implement the key features of the “Watch ‘n’ Buy” system, i.e. user management, annotation management, data import, semantics extraction, metadata enrichment, etc. The Interface Layer (L3) is the entry point from which the users and third-party applications will be able to interact with the system. The rest of this section details the core components of the architecture.

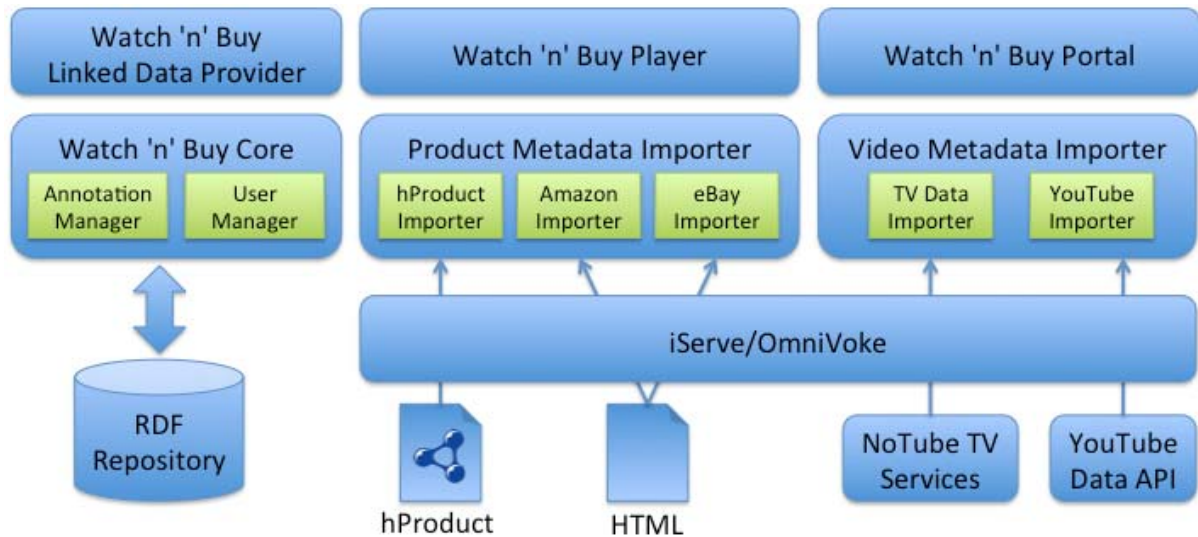


Figure 3. Software components of “Watch ‘n’ Buy”

4.2.1. RDF Data Repository

All the metadata of annotations, video clips and products are stored in a RDF repository developed based on OpenRDF Sesame³³ and OWLim³⁴. It also exposes a standard compliant SPARQL endpoint with the new features of SPARQL 1.1³⁵, e.g. data aggregation, sub-query, negation, etc.

4.2.2. Video Metadata Importer

As illustrated in Figure 3, for each video sharing website or TV channel, a Linked Service will be developed and registered in iServe. When a user is adding annotations to a video clip or TV programme, the metadata will be imported via invoking the corresponding Linked Service.

4.2.3. Product Metadata Importer

Similar to the video metadata importer, two Linked Services will be developed for cropping metadata of products. One is dedicated to processing the HTML pages having the micro-format of hProduct. The other one is a generic service of transforming HTML pages into RDF triples. The idea was inspired by a website called ScraperWiki³⁶, which aims to enable the sharing of “screen scrapers” in a wiki-like way. A screen scraper is a piece of codes that can extract useful data from Web pages. An inherent nature of screen scrapers is the high reusability, which provides the possibility of the aforementioned crowdsourcing of product metadata harvesting. With the generic transformation service, the “Watch ‘n’ Buy” system can query and execute screen scrapers to generate RDF statements. Third-party developers are allowed to upload new screen scrapers to extend the capability of the transformation service.

4.2.4. Watch ‘n’ Buy Player

The key features of “Watch ‘n’ Buy” player are as follows:

- Adding annotations to a video clip with narrative descriptions or URLs of products
- Specifying the temporal property of an annotation, which can be a time point or a period of time
- Specifying the spatial property of an annotation, which can occupy a region of the screen
- Showing the existing annotations when a video is pause
- Gathering users’ feedbacks on annotations, e.g. reporting irrelevant annotations, refining the position and adding comments

³³ <http://www.openrdf.org/>

³⁴ <http://www.ontotext.com/owlim>

³⁵ <http://www.w3.org/TR/sparql11-query/>

³⁶ <https://scraperwiki.com/>

Two versions of the “Watch ‘n’ Buy” player, i.e. a Web-based one and a second-screen-based one will be developed for two different scenarios: online video watching and TV programme watching. The Web based “Watch ‘n’ Buy” player will be implemented by using and extending the JavaScript API of online video stream player such as YouTube player API³⁷. The second-screen based one will be realised by making use of the TEAPlayer developed in working package 7c of the NoTube project (see D7c.3).

4.2.5. Watch ‘n’ Buy Portal

The “Watch ‘n’ Buy” portal is responsible for providing the essential functionalities of Web portal, e.g. browsing videos and products, filtering by categories and tags, searching by key words, etc. It also provides social networking functionalities such as followers and feeds. The objective of “Watch ‘n’ Buy” portal is to set up an online community to encourage annotating video clips and TV programmes with semantically enhanced descriptions of products.

4.2.6. Linked Data provider

The Linked Data provider is a front-end tool of publishing Linked Data on the Web. Generally, it performs two tasks: content negotiation and data serialisation. Content negotiation is a mechanism of determining the format of the response by analyzing the headers of an HTTP request. From the perspective of a client, the “Accept” header can be used to require the system to serialize the data in a desirable format.

The data serialisation component can be divided into two parts: one is in charge of serialising metadata as human-readable HTML pages. For the ease of consumption by third-party applications, the other part of the data serialisation component can serialise metadata using different RDF syntaxes such as XML, TriG, TriX, N-Triples, N3, etc.

4.2.7. Other Components

Here we briefly describe two components that are not included in the architecture diagram.

Bookmarklet

A bookmarklet, called “Watch ‘n’ Buy It”, will be developed for the loosely-coupled integration of “Watch ‘n’ Buy” and the video sharing websites like YouTube. When clicked, the bookmarklet will bring the user to the “Watch ‘n’ Buy” player so that the user can have access to the annotations associated to the video. Essentially, the bookmarklet is a piece of JavaScript codes can be stored as a bookmark in the Web browser.

Web browser plug-in

Several plug-ins will be developed for major browsers such as Chrome, Firefox and Internet Explorer. They aim to provide the following functionalities:

- When a user is watching a video, the plug-in tells how many annotations have been attached to the video
- When opened, the plug-in shows a list of products that appear in the video
- The plug-in allows filtering the annotations by key words
- The plug-in can redirect the user to the eCommerce websites selling those products

³⁷ http://code.google.com/apis/youtube/js_api_reference.html

5. External Interfaces

This section describes the interfaces of “Watch ‘n’ Buy”, covering not only the GUI for end-users and third-party developers, but also the RESTful API and SPARQL endpoint for machines. Mockups are shown in this section to visualise the GUI of the video player and the portal. Technical details of the RESTful API, e.g. parameters, data schema of invocation results, are also specified in this section.

5.1. Watch ‘n’ Buy Player

As mentioned before, there will be two versions of the “Watch ‘n’ Buy” player: one for Web browser, the other one for the second screen. In this sub-section, we introduce respectively their user interfaces.

5.1.1. Web based User Interface

Figure 4 presents the mockup of Web based “Watch ‘n’ Buy” player. On the left hand side, a video player performs the playback functionality of programmes as normal. When pause, annotations (denoted as flags) will appear on the screen. Meanwhile, a list of products will be shown on the right hand side. Clicking on somewhere on the video, the users can add new annotations using the pop-up dialog.

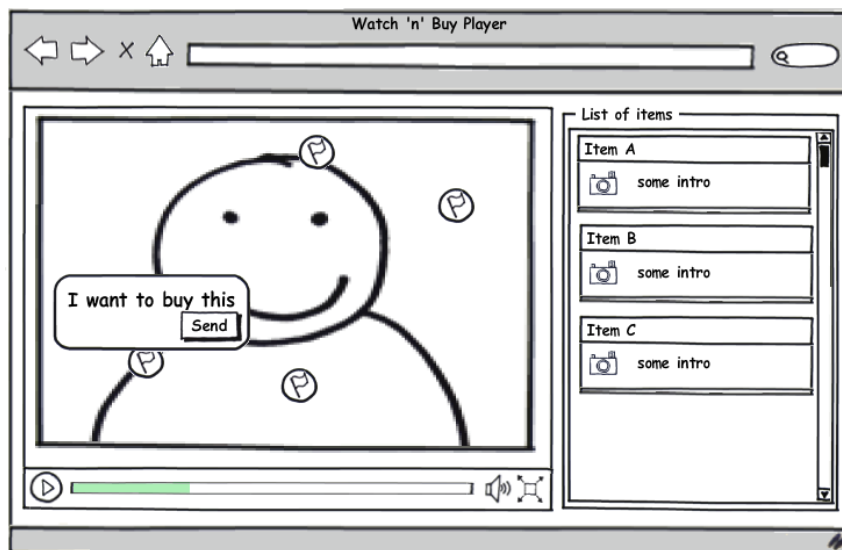


Figure 4. Mockup of Web-based “Watch ‘n’ Buy” player

5.1.2. User Interface on the Second Screen

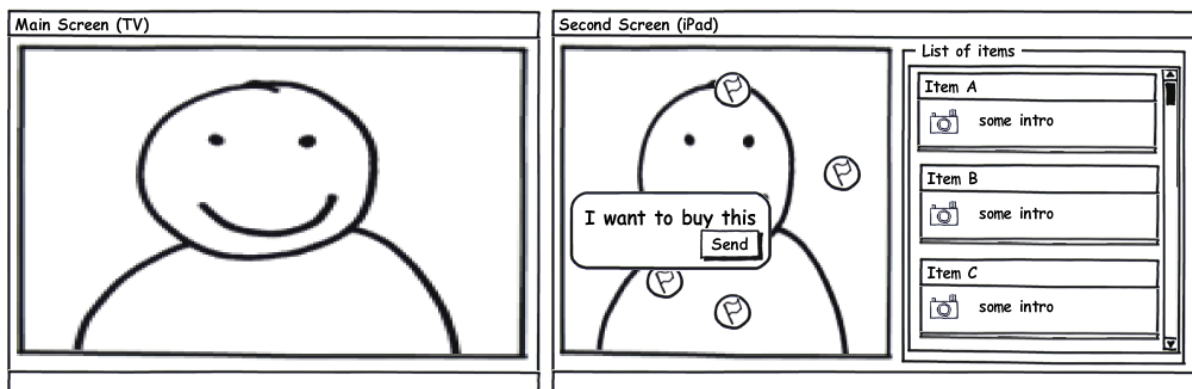


Figure 5. Mockup of “Watch ‘n’ Buy” player on the second screen

Figure 5 shows the mockup of “Watch ‘n’ Buy” player based on the second screen. Because it is difficult to manipulate the main screen, especially for TV and movie, the annotation has to be done on the second screen. In this case, it is required that every TV programme must have a dereferenceable URI. Otherwise, the second screen cannot be synchronized with the main screen. Taking into account the fact that there may not be metadata describing certain TV programmes, nor services for retrieving screenshots, the second-screen based player may have less functionalities than the browser-based one.

5.2. Watch ‘n’ Buy Portal

Figure 6 depicts the mockup of the “Watch ‘n’ Buy” portal, which is a typical social networking website. A timeline of the annotations made by a user’s friends is automatically generated and displayed on the left hand side. Each item of the timeline shows information like: “User1 annotated Video1 with Product1”. It also includes screenshot of the video as well as an image of the product. In addition, the items of the timeline can tell the provenance of the annotation.

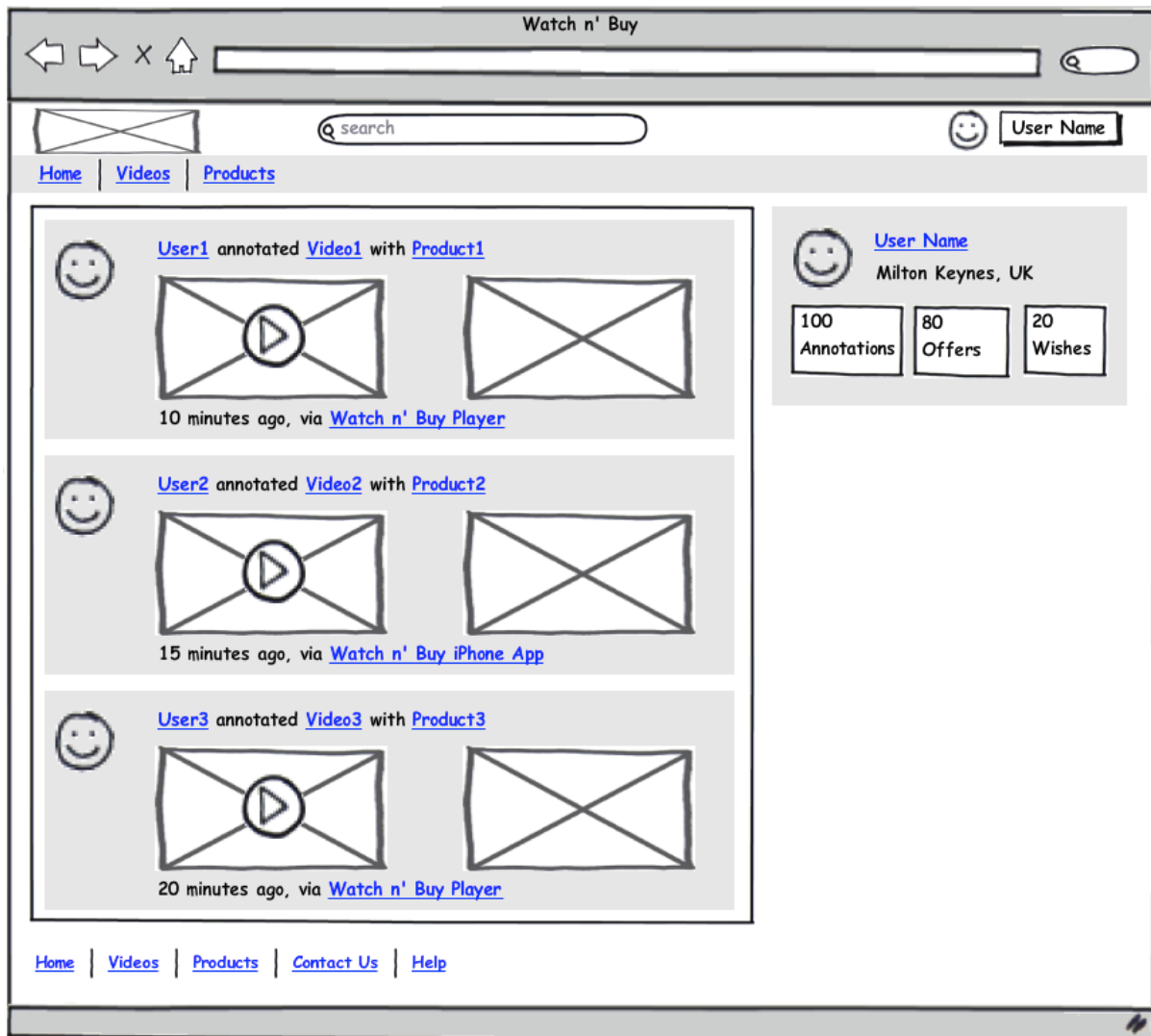


Figure 6. Mockup of “Watch ‘n’ Buy” portal

5.3. RESTful API

This sub-section specifies the core functions of the RESTful API of “Watch ‘n’ Buy”. In short, it covers part of the CRUD (Create, Read, Update and Delete) operations on users, annotations, products, videos, ratings, comments, etc. In the rest of this sub-section, we detail the input parameters of each function, as well as the data schema of the outputs.

5.3.1. User

Retrieve the profile of a user

HTTP Method: GET

Requires Authentication: False

URL: /api/user/{id}

Parameters:

| Name | Type | Required/Optional | Description |
|------|--------|-------------------|---|
| id | String | Required | The identifier of the user, which can be either a UUID or the user name |

Response:

| Name | Type | Description |
|-------------------|---------|---|
| id | String | UUID of the user |
| name | String | Name of the user |
| description | String | Introduction of the user |
| annotations_count | Integer | Number of annotations the user has made |
| followers_count | Integer | Number of the followers of the user |

Follow a user

HTTP Method: POST

Requires Authentication: True

URL: /api/follow/create

Parameters:

| Name | Type | Required/Optional | Description |
|-----------|--------|-------------------|--|
| target_id | String | Required | The identifier of the target user, which can be either a UUID or the user name |

Response:

If successful, this function returns information about the target user, using the same schema as the one for retrieving a user profile.

5.3.2. Annotation

Retrieve an annotation

Method: GET

Requires Authentication: False

URL: /api/annotation/{id}

Parameters:

| Name | Type | Required/Optional | Description |
|------|--------|-------------------|------------------------|
| id | String | Required | UUID of the annotation |

Response:

| Name | Type | Description |
|---------------|-----------|---|
| id | String | UUID of the annotation |
| url | URL | URL of the annotation |
| description | String | Description of the annotation |
| annotator_id | String | UUID of the user who created the annotation |
| video_id | String | UUID of the annotated video |
| product_id | String | UUID of the product used to make the annotation |
| mood | Enum | Seek, Saw or Sell |
| start_time | Integer | Starting time of the annotation in millisecond |
| end_time | Integer | End time of the annotation in millisecond |
| x | Float | Normalised coordinate on x axis |
| y | Float | Normalised coordinate on y axis |
| width | Float | Normalised width of region |
| height | Float | Normalised height of region |
| creation_time | Date time | The time at which the annotation was made |

Create an annotation

Method: POST

Requires Authentication: True

URL: /api/annotation/create

Parameters:

| Name | Type | Required/Optional | Description |
|-------------|---------|-------------------|---|
| video_url | URL | Required | The URL of the video to be annotated |
| product_url | URL | Required | The URL of the product to be used to annotate the video |
| description | String | Optional | Description of the annotation |
| mood | Enum | Required | Seek, Saw or Sell |
| start_time | Integer | Required | Starting time of the annotation in millisecond |
| end_time | Integer | Required | End time of the annotation in millisecond |
| x | Float | Required | Normalised coordinate on x axis |
| y | Float | Required | Normalised coordinate on y axis |
| width | Float | Optional | Normalised width of region |
| height | Float | Optional | Normalised height of region |

Response:

If successful, this function returns information about the created annotation, using the same schema as the one for retrieving an annotation.

Search for annotations

Method: GET

Requires Authentication: False

URL: /api/annotation/search

Parameters:

| Name | Type | Required/Optional | Description |
|------------|--------|-------------------|--|
| video_id | String | Optional | The identifier of the video to be used as the parameter of searching, i.e. searching for annotations attached to the given video |
| product_id | String | Optional | The identifier of the product to be used as the parameter of searching |
| user_id | String | Optional | The identifier of the user to be used as the parameter of searching |

Response:

This function returns a list of the UUID of relevant annotations.

5.3.3. Product

Retrieve a product

Method: GET

Requires Authentication: False

URL: /api/product/{id}

Parameters:

| Name | Type | Required/Optional | Description |
|------|--------|-------------------|---------------------|
| id | String | Required | UUID of the product |

Response:

| Name | Type | Description |
|------------------|---------|--|
| id | String | UUID of the product |
| url | URL | URL of the product |
| title | String | A short name of the product |
| description | String | Detailed description of the product |
| img | URL | URL of the image of the product |
| annotation_count | Integer | Number of the annotations that use the product |
| offer_count | Integer | Number of the offers of the product |

Retrieve an offer

Method: GET

Requires Authentication: False

URL: /api/offer/{id}

Parameters:

| Name | Type | Required/Optional | Description |
|------|--------|-------------------|---------------------|
| id | String | Required | UUID of the product |

Response:

| Name | Type | Description |
|-------------|--------|---|
| id | String | UUID of the offer |
| url | URL | URL of the offer |
| product_id | String | UUID of the offering product |
| vendor_id | String | UUID of the vendor |
| description | String | Detailed description of the offer |
| origin_url | URL | Corresponding URL at the vendor's website |
| price | String | Descriptions of the price |

Create a metadata importer

Method: POST

Requires Authentication: True

URL: /api/importer/create

Parameters:

| Name | Type | Required/Optional | Description |
|----------------|--------|-------------------|--|
| name | String | Required | A short name of the data importer |
| target_website | URL | Required | URL of the target website |
| format | Enum | Required | XSLT, XSPARQL or JRON |
| content | String | Required | The content of the data importer, e.g. the XSLT specification |
| example_url | URL | Optional | URL of an example for testing and validation of the transformation |

Response:

If successful, this function returns the URL of the importer.

5.3.4. Video

Retrieve a video

Method: GET

Requires Authentication: False

URL: /api/video/{id}

Parameters:

| Name | Type | Required/Optional | Description |
|------|--------|-------------------|-------------------|
| id | String | Required | UUID of the video |

Response:

| Name | Type | Description |
|------------------|---------|--|
| id | String | UUID of the video |
| url | URL | URL of the video |
| title | String | A short name of the video |
| description | String | Detailed description of the video |
| img | URL | Screenshot of the video |
| annotation_count | Integer | Number of the annotations on the video |
| duration | Integer | Duration of the video in seconds |

5.3.5. Comment

Retrieve a comment

Method: GET
Requires Authentication: False
URL: /api/comment/{id}
Parameters:

| Name | Type | Required/Optional | Description |
|------|--------|-------------------|---------------------|
| id | String | Required | UUID of the comment |

Response:

| Name | Type | Description |
|---------------|-----------|--|
| id | String | UUID of the comment |
| url | URL | URL of the comment |
| item_url | URL | URL of the item being commented |
| reviewer_id | String | UUID of the reviewer |
| comment | String | Content of the comment |
| creation_time | Data time | The time at which the comment was made |

Create a comment

Method: POST
Requires Authentication: True
URL: /api/comment/create
Parameters:

| Name | Type | Required/Optional | Description |
|----------|--------|-------------------|---------------------------------|
| item_url | URL | Required | URL of the item being commented |
| comment | String | Required | Content of the comment |

Response:

If successful, this function returns information about the created comment, using the same schema as the one for retrieving a comment.

5.3.6. Rating

Retrieve a rating

Method: GET
Requires Authentication: False
URL: /api/rating/{id}
Parameters:

| Name | Type | Required/Optional | Description |
|------|--------|-------------------|--------------------|
| id | String | Required | UUID of the rating |

Response:

| Name | Type | Description |
|---------------|-----------|---------------------------------------|
| id | String | UUID of the rating |
| url | URL | URL of the rating |
| item_url | URL | URL of the item being rated |
| reviewer_id | String | UUID of the reviewer |
| rating | String | The value of the rating |
| min | String | The minimum of the rating |
| max | String | The maximum of the rating |
| creation_time | Data time | The time at which the rating was made |

Create a rating

Method: POST
Requires Authentication: True
URL: /api/rating/create
Parameters:

| Name | Type | Required/Optional | Description |
|----------|--------|-------------------|-----------------------------|
| item_url | URL | Required | URL of the item being rated |
| rating | String | Required | The value of the rating |

| | | | |
|-----|--------|----------|---------------------------|
| min | String | Required | The minimum of the rating |
| max | String | Required | The maximum of the rating |

Response:

If successful, this function returns information about the created rating, using the same schema as the one for retrieving a rating.

5.3.7. Tag

Retrieve a tag

Method: GET

Requires Authentication: False

URL: /api/tag/{id}

Parameters:

| Name | Type | Required/Optional | Description |
|------|--------|-------------------|-----------------|
| id | String | Required | UUID of the tag |

Response:

| Name | Type | Description |
|---------------|-----------|--|
| id | String | UUID of the tag |
| url | URL | URL of the tag |
| item_url | URL | URL of the item being tagged |
| reviewer_id | String | UUID of the reviewer |
| name | String | A short name of tag |
| url | URL | URL of a dereferenceable resource on the Web of Data |
| creation_time | Data time | The time at which the rating was made |

Create a tag

Method: POST

Requires Authentication: True

URL: /api/tag/create

Parameters:

| Name | Type | Required/Optional | Description |
|----------|--------|-------------------|--|
| item_url | URL | Required | URL of the item being tagged |
| name | String | Required | A short name of tag |
| url | URL | Required | URL of a dereferenceable resource on the Web of Data |

Response:

If successful, this function returns information about the created tag, using the same schema as the one for retrieving a tag.

5.4. SPARQL Endpoint

For the ease of interaction with third-party applications, “Watch 'n' Buy” will expose a SPARQL endpoint. It accepts HTTP requests and sends back the serialisation of query results in the format of standardised XML³⁸.

³⁸ <http://www.w3.org/TR/rdf-sparql-XMLres/>

6. Related Work

In this section, we highlight the work related to eCommerce-oriented video annotation, inspired by which, “Watch ‘n’ Buy” is proposed.

6.1. Annomation

Annomation³⁹ (see Figure 7) is a semantic video annotation platform for annotating and publishing educational video resources using Linked Data identifiers and free text. Essentially, it is a Web 3.0 application that allows users to collaboratively view a video, pause it, and add Linked Data annotations to instants or durations in the video.

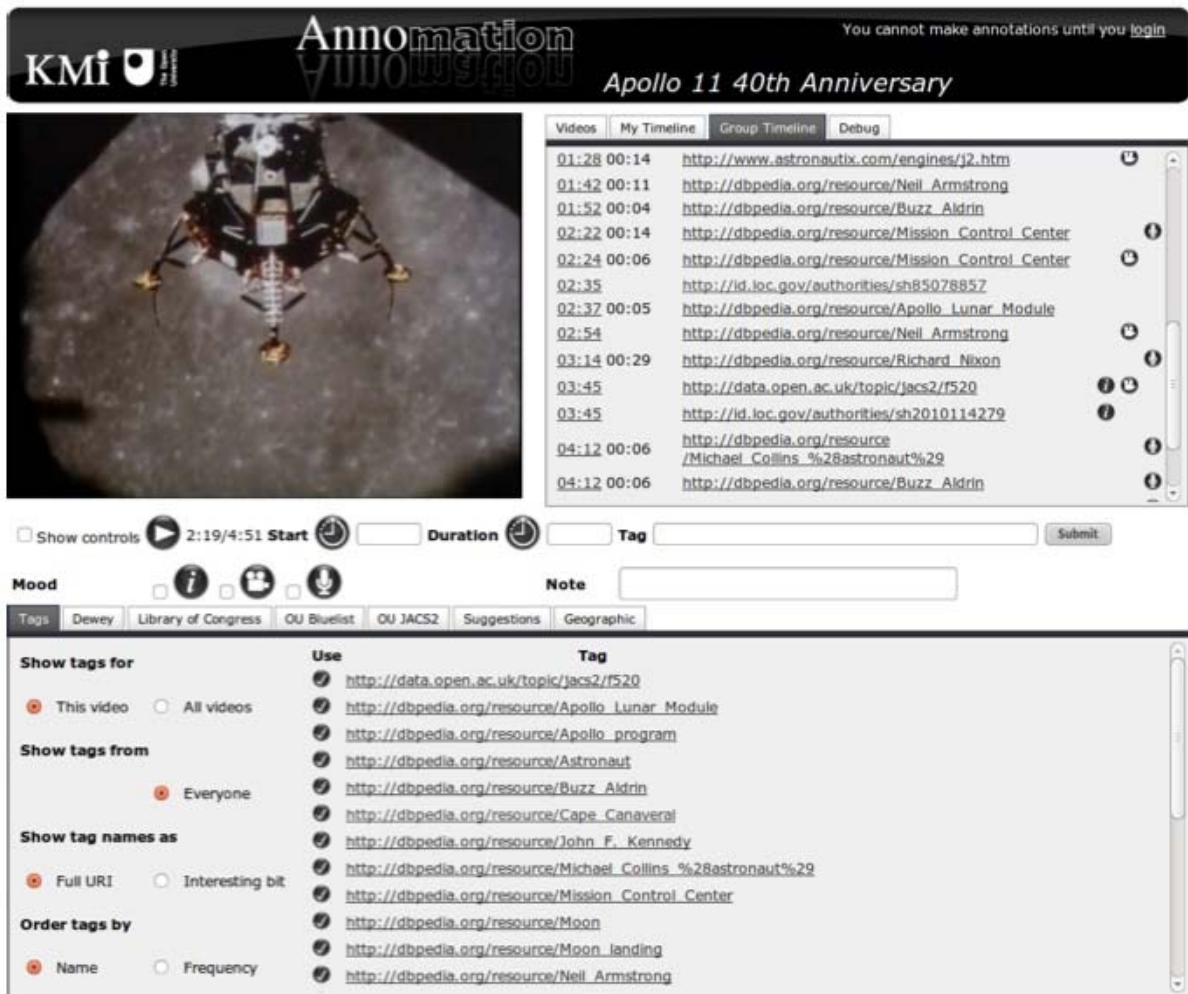


Figure 7. Screenshot of Annomation

Five Linked Services have been used as the foundations to annotate videos and they are embedded in the Annomation functions to facilitate the annotation process. They are:

- **Dewey Decimal:** The top level Dewey Decimal Classification (covering the first three digits of a Dewey number) has been published in RDF form by the Online Computer Library Centre⁴⁰, and the resulting taxonomy is presented to the annotator as a browsable tree.
- **Library of Congress classifications:** The Library of Congress has published its entire classification system in RDF⁴¹.

³⁹ <http://annomation.open.ac.uk/annomation>

⁴⁰ <http://dewey.info/>

⁴¹ <http://id.loc.gov/authorities/>

- **GeoNames:** The GeoNames⁴² API is used to identify named locations according using keyword search, or perform reverse lookup to find named locations in a vicinity.
- **OU Bluelist:** This is a service can be used to get Open University course taxonomies.
- **Zemanta:** A service aimed at helping bloggers augment their posts with links and images, Zemanta⁴³ analyses natural language text to identify various concepts and named entities, returning URIs to Linked Data.

Annotation is one of the results of our previous work. Same as “Watch ‘n’ Buy”, it is also built following Linked Services principles. Moreover, we extend the conceptual model of Annotation to support the eCommerce-oriented video annotation, especially capturing both the temporal and spatial properties of annotations.

6.2. SugarTube

SugarTube⁴⁴ (see Figure 8) is a semantic-based video searching browser for searching videos annotated by Annotation and providing more linked videos and educational resources from the Web. It adopts the Linked Data approach to search videos and explore their related online resources in a mashup navigation interface. In SugarTube, the annotations are semantically matched to other annotated educational resources from the Web.

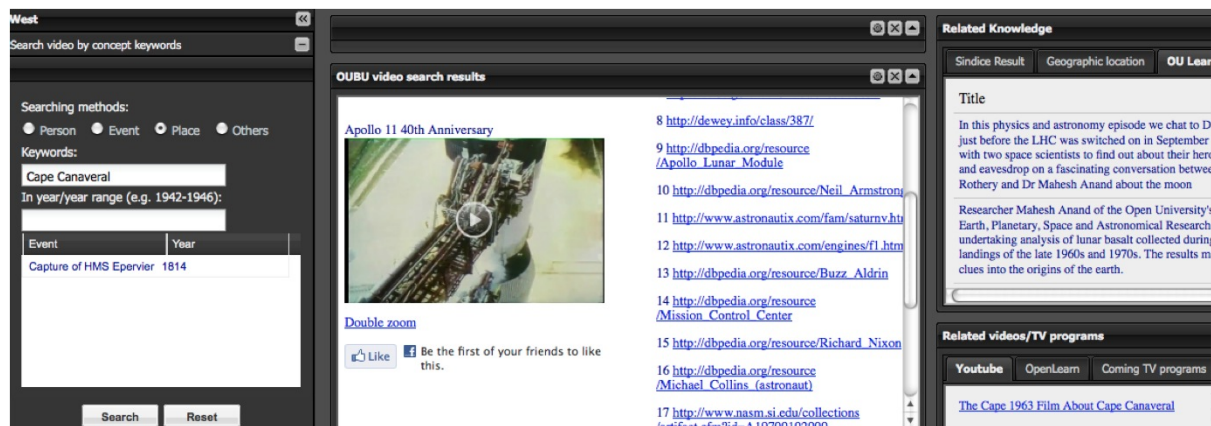


Figure 8. Screenshot of SugarTube

The Web-based video player of “Watch ‘n’ Buy” is implemented re-using part of the source codes of SargarTube. However, “Watch ‘n’ Buy” player presents the annotations in a more intuitive way, i.e. show the annotations as markers on the screen. We expect the user experience to be like grabbing things directly from the screen.

6.3. Things You Saw in a Movie

“Things You Saw in a Movie”⁴⁵ is a website, where the links to the products of the stars in movies can be found. This website can be regarded as a step towards implementation of WYSIWYCB (What You See Is What You Can Buy), for the reasons: 1) it provides direct links to product vendors, e.g. Amazon.com; 2) it presents screenshots of movies and the images of products. However, it does not implement the functionality of video annotation and playback. And, the website seems to be a blog rather than an online community.

6.4. Tesco’s Billboard Grocery Store

The Tesco’s Billboard Grocery Store⁴⁶ is a system that allows people buying products shown on a poster in a subway station. What a customer need to do is scanning the QR code of a product and making the payment with a smartphone. The product will be delivered to the specified address. This is

⁴² <http://www.geonames.org/>

⁴³ <http://www.zamanta.com/>

⁴⁴ <http://sugartube.open.ac.uk/>

⁴⁵ <http://www.thingsyousawinamovie.com/>

⁴⁶ <http://iamkoream.com/tescos-billboard-grocery-store-a-hit-in-s-korea>

a remarkable attempt to attach links to visible objects for the ease of purchase. However, this system does not intend to make purchasable the products appearing in video. And, it does not allow end-users adding new links, which is a significant difference from the vision of “Watch ‘n’ Buy”.

7. Conclusion

In this deliverable we propose the motivations, requirements and the design of “Watch ‘n’ Buy”, an eCommerce-oriented video annotation environment. By allowing both product vendors and customers annotating video clips, it will implement a novel advertisement insertion model. On the other hand, the image recognition, especially for the products shown in video, can be performed by utilizing the power of crowd.

We plan to develop “Watch ‘n’ Buy” as a social semantic Web application. Therefore, we use ontologies and vocabularies to represent the metadata of annotations, videos and products. And, the architecture of “Watch ‘n’ Buy” is designed following the paradigm of Linked Services. The implementation and usage of “Watch ‘n’ Buy will be described in upcoming deliverables D5.5.

8. References

- [1] Hepp, M. (2008). GoodRelations: An Ontology for Describing Products and Services Offers on the Web. In Proceedings of 16th International Conference on Knowledge Engineering and Knowledge Management Knowledge Patterns (EKAW 2008). Gangemi, A. and Euzenat J. eds. LNCS 5268, pp. 329–346.
- [2] Pedrinaci, C., Liu, D., Maleshkova, M., Lambert, D., Kopecky, J., and Domingue, J. (2010). iServe: a Linked Services Publishing Platform, Workshop: Ontology Repositories and Editors for the Semantic Web at 7th Extended Semantic Web Conference.
- [3] Li, N., Pedrinaci, C., Maleshkova, M., Kopecky, J., Domingue, J. (2011). OmniVoke: A Framework for Automating the Invocation of Web APIs. In Proceedings of 5th IEEE International Conference on Semantic Computing, Palo Alto, USA.

A. Appendix A

This document aims to clarify the adjustment of work plan in WP5 of NoTube project. Essentially, the major efforts of WP5 are going to be devoted to design and implement a use case called "Watch 'n' Buy" since M31, instead of focusing on context-aware discovery and mediation for media resources based on WSMO. "Watch 'n' Buy" aims to enable both consumers and manufactures annotating online video and TV programs with semantic descriptions of products. In this way, advertisements attached to video and TV program will become interactive and less-disturbing.

1. Motivation

Advertisements are now ubiquitous but are not interactive. For instance, when a person is interested in some soft drink on TV, how can s/he buy one? S/he needs to search for a shop selling the soft drink, which might be difficult and time consuming. Also it is not desirable that a customer or consumer should have to take responsibility for finding products that s/he wants to buy.

On the other hand, advertisements attached to TV programs or online videos are disturbing to a certain extent. One of the reasons is that those advertisements are always visible to all the audience, and are played sequentially, before or after TV programs. We can improve the user experience if the advertisements are 'played' in parallel with programs, and only become visible at a user request.

In addition, explicit or implicit celebrity endorsement is a very efficient advertising channel. For example, Kate Middleton's Royal Wedding Ring is now a very popular gift in the UK. If both consumers and manufactures are able to annotate TV programs and video, it will be easy for them to establish connections with each other.

This application, "Watch 'n' Buy", aims to address these issues. Our ultimate goal is to implement WYSIWCB: What You See (on TV, in online video, in poster, etc.) Is What You Can Buy. We will first focus on online video and TV programs.

From all above, the reason for changing the work plan of WP5 is that we believe "Watch 'n' Buy" has potential impacts on both broadcasting media delivery and the advertising industry.

2. Updated Work Plan

According to the original work plan of WP5, the remaining work includes two tasks, i.e. T5.3 (Context-aware discovery and mediation for media resources based on WSMO) and T5.4 (Implementation and Integration of SWS-based TV Resource Broker). The former intends not only to extend WSMO ontology with facilities enabling matchmaking between real-world context parameters and SWS representations, but also to implement fuzzy mediation for WSMO services. The latter aims at integrating the outcomes of the other tasks in WP5.

The tasks in WP5 are re-defined as follows.

Task 5.3. Discovery, Invocation and Integration of TV Services. This task deals with issues related to T5.3 and T5.4 of the original work plan. First, this task is going to finalize the semantic annotations of services provided by partners. Moreover, this task also aims at refining the conceptual model of the non-functional properties of services. After that, data about the non-functional properties of services will be gathered and published as Linked Data. On the other hand, this task involves developing a service execution engine. Overall, the target of this task is to make the TV services discoverable and invocable.

Task 5.4. Video Annotation Tool Design. T5.4 involves design of the aforementioned application "Watch 'n' Buy". First, a data schema for annotating video streams will be defined, which will reuse the existing ontologies, e.g. FOAF⁴⁷, GoodRelations⁴⁸, Time Ontology⁴⁹, Timeline Ontology⁵⁰. Second, the overall architecture of "Watch 'n' Buy" will be designed, following the Service-Oriented Architecture (SOA) paradigm. "Watch 'n' Buy" will be built on top of semantic repositories that store annotations, user profiles and provenance information. It will provide a Web based player facilitating users annotating online video. Furthermore, in order to enable annotating TV programs, "Watch 'n' Buy" will invoke the Web services for retrieving metadata, and

⁴⁷ <http://xmlns.com/foaf/spec/>

⁴⁸ <http://www.heppnetz.de/ontologies/goodrelations/v1.html>

⁴⁹ <http://www.w3.org/TR/owl-time/>

⁵⁰ <http://motools.sourceforge.net/timeline/timeline.html>

also make use of the Second Screen techniques. Additionally, “Watch ‘n’ Buy” will publish the annotations on video stream as Linked Data, and expose a SPARQL endpoint as well as Web APIs.

Task 5.5. Video Annotation Tool Implementation. This task includes the development of “Watch ‘n’ Buy”. In general, it will be implemented with Semantic Web Services, and realise the following functionalities: annotating online video and TV programs, searching for existing annotations, rating and reviewing on annotations, semantics extraction from text descriptions of user inputs, etc.

According to the updated work plan, WP5 will produce the following deliverables:

D5.3 Discovery, Invocation and Integration of TV Services (M33)

D5.3 will summarise the work related to the original work plan of WP5. In particular, it will elaborate the discovery mechanism of TV services and the generic Web service invocation engine. This deliverable will also include the software of invocation engine.

D5.4 Design of video stream annotation environment (M33)

“Watch ‘n’ Buy”, the environment for annotating video streams with semantic descriptions of products, is to be designed in this deliverable. As stated before, the design of “Watch ‘n’ Buy” will follow the SOA paradigm, and the core functionalities will be encapsulated as RESTful Web services. D5.4 will document the formal design specification, which includes the definition of adopted data schemas and the layered system architecture. In addition, D5.4 will also clarify the decisions on underlying components, such as RDF repositories, RESTful service framework, etc.

D5.5 “Watch ‘n’ Buy” prototype (M36)

D5.5 will provide prototypes of “Watch ‘n’ Buy”, based on the design specification delivered in D5.4. The development of “Watch ‘n’ Buy” will be conducted in an agile and iterative way. The first prototype will be a proof of concept, and more advanced and mature versions will be produced through several iterations.

3. Resources

The design and development of “Watch ‘n’ Buy” will take advantage of our previous work on SugarTube⁵¹ and Annomation⁵², and also outcomes of other WPs, e.g. definition of metadata for TV programs and Web services for retrieving and those metadata.

In respect of human resources, according to the estimate of the workloads of both the remaining tasks and “Watch ‘n’ Buy”, we plan to spend 6 person-months to the work of “Watch ‘n’ Buy”.

4. Impact

The updated work of plan of WP5 will make remarkable contributions to WP7 (User Cases) and WP8 (Show Cases). On the other hand, it has little negative influence on the tasks related to the outcomes of WP5, because software supporting to build the TV broker, i.e. SmartLink, iServe and IRS-III have already been available and fairly mature.

⁵¹ <http://sugartube.open.ac.uk/>

⁵² <http://annomation.open.ac.uk/annomation>