vista TV

SEVENTH FRAMEWORK PROGRAMME

**ViSTA-TV: Video Stream Analytics for Viewers in the TV Industry**

FP7 STREP ICT-296126 | 296126 co-funded by the European Commission
ICT-2011-SME-DCL | SME Initiative on Digital Content and Languages

# D6.2 External data service deployed

Valentina Maccatrozzo (VUA),
Lora Aroyo (VUA),
Guus Schreiber (VUA),
Nils Wöhler (Rapid-i),
Simon Fischer (Rapid-i).

| | | | |
|---|---|---|---|
| Project start date: | June $1^{st}$, 2012 | Project duration: | 24 months |
| Document identifier: | ViSTA-TV/2012 D6.1 | Version: | v1.0 |
| Date due: | 01 June 2013 | Status: | Final version |
| Submission date: | 17 May 2013 | Distribution: | RE |

**www.vista-tv.eu**

# ViSTA-TV Consortium

This document is part of a collaborative research project funded by the FP7 ICT Programme of the Commission of the European Communities, grant number 296126. The following partners are involved in the project:

**University of Zurich (UZH) - Coordinator**
Dynamic and Distributed Information Systems Group (DDIS)
Binzmühlstrasse 14
8050 Zürich, Switzerland
Contact person: Abraham Bernstein
E-mail: bernstein@ifi.uzh.ch

**Techniche Universität Dortmund (TUDo)**
Computer Science VIII: Artificial Intelligence Unit
D-44221 Dortmund, Germany
Contact person: Katharina Morik
E-mail: katharina.morik@cs.uni-dortmund.de

**Rapid-I GmbH (RAPID-I)**
Stockumer Strasse 475
44227 Dortmund, Germany
Contact person: Ingo Mierswa
E-mail: mierswa@rapid-i.com

**Zattoo Europa AG (Zattoo)**
Eggbühlstrasse 28
CH-8050 Zürich, Switzerland
Contact person: Bea Knecht
E-mail: beaknecht@me.com

**Vrije Universiteit Amsterdam (VUA)**
Web & Media Group, Department of Computer Science, Faculty of Sciences (FEW)
De Boelelaan 1081a
NL-1081 HV Amsterdam, The Netherlands
Contact person: Guus Schreiber
E-mail: guus.schreiber@vu.nl

**The British Broadcasting Corporation (BBC)**
56 Wood Lane / Centre House - Broadcasting House
UK-W12 7SB Northampton, United Kingdom
Contact person: Chris Newell
E-mail: Chris.Newell@bbc.co.uk

# Executive Overview

This document describes the external data service deployment. In particular, it describes in details the technical aspects of the service: how, how often and where the data is put at disposal to all the partners. It regards not only the external data, but also the Electronic Program Guide (EPG) data provided by our broadcaster partners.

# Contents

# 1 Introduction

During the first year of the project, WP6 worked on the implementation of the external data service. This includes not only the implementation of the workflow used, but also the provision of the EPG data in the data warehouse. This document explains in details the whole process as well as future implementations. The external data service has the main objective of adding useful information to the EPG data made available by our broadcaster partners, in order to improve the recommendations.

In Section 2 we present an overview of the workflow, in Section 3 we introduce the external data engine. In Section 4, we describe the procedure to populate the Data Warehouse, while in Section 5 we present the API developed to query the Data Warehouse. Finally in Section 6 we illustrate the additional features we intend to provide during the next year.

# 2 The workflow

WP6 makes available to the other partners the data used in the project. We are responsible for populating the data warehouse with the EPG data and with the related external data. The data warehouse provides data up to seven days in advance. This procedure is complex, but it can be summarized into two big components:

1. retrieve EPG data from the broadcasters, and load it into the data warehouse
2. perform the enrichment of the EPG data, and load it into the data warehouse

These two steps are performed at the same time. The coming sections describe in detail the two steps in order of importance.

# 3 External data engine

In this section we present the procedure for the linking of the EPG data to external data sources. The workflow is similar for the two broadcasters' data. We use Resource Description Framework (RDF) representation of the data in order to perform the enrichments. External data is provided for the title of the program, the people involved in the program and for the synopsis. For the latter we use an external service to extract concept from the text, which are matched with Linked Open Data (LOD) entities. For the moment we use only DBpedia, and in particular we provide DBpedia categories for programs, credits and synopsis concepts. The workflow is composed by the following steps:

- retrieve the program RDF file
- load it into an RDF store
- look for LOD URIs for title and credits
- perform a concept extraction from the text of the synopsis
- look for LOD URIs for these concepts
- extract labels for the LOD entities and store them into the data warehouse.

As already said the procedure differs slightly for the two broadcasters. We explain it in details in the following sections.

## 3.1 BBC

BBC EPG data is already available in RDF format, so we do not need to perform any transformation of the data format. We load the data into an RDF store, and we perform the enrichment, as described before.

In particular, the concept tagging step is performed with AlchemyAPI[1] Concept Tagging API[2]. The input for this service is the text of the synopsis, and the output is in RDF format, where the identified concepts are matched with LOD entities.

[1] http://www.alchemyapi.com
[2] http://www.alchemyapi.com/api/concept

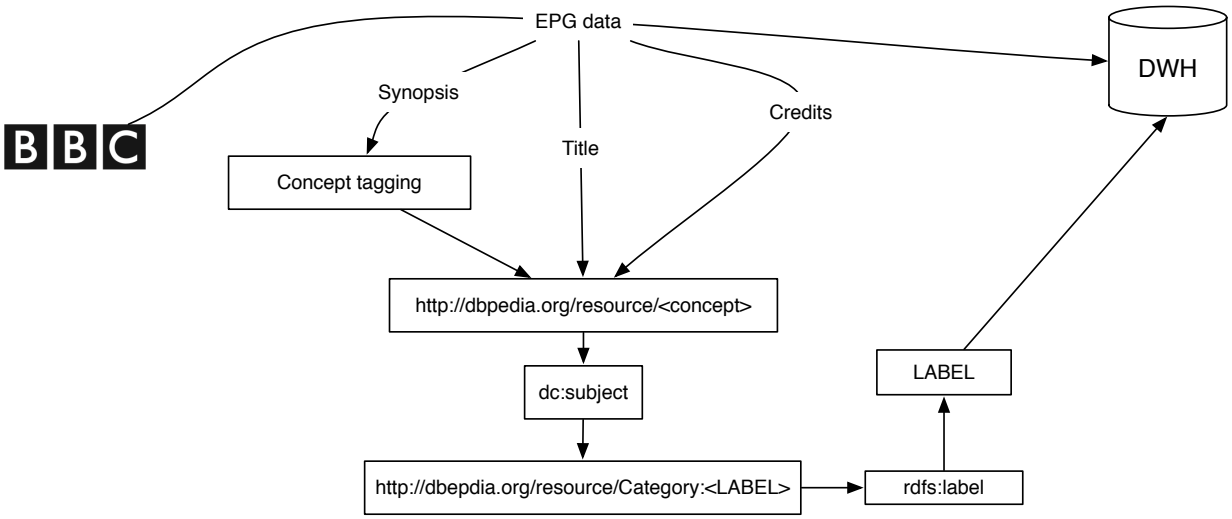


Figure 1: BBC workflow

## 3.2 Zattoo

Zattoo data is not available in RDF format, so we retrieve the data in XML format and we translate it into RDF. The schema used is the one provided by the Programmes Ontology[3], the same used by BBC (see Section 3.2.1 and Appendix C for the ontology specifications). Once the file is created, we load into the RDF store, and the workflow follows the above description. Besides, since AlchemyAPI provides the Concept Tagging API only for English, we use a different service for the concept tagging step, which is Lupedia[4]. Lupedia is a service developed in the context of the NoTube Project[5], which can deal with the additional languages we have. The language is an input field, hence we need to apply a language detection procedure for the extraction of the concepts from the synopsis. This last step is performed with AlchemyAPI Language Detection API[6].

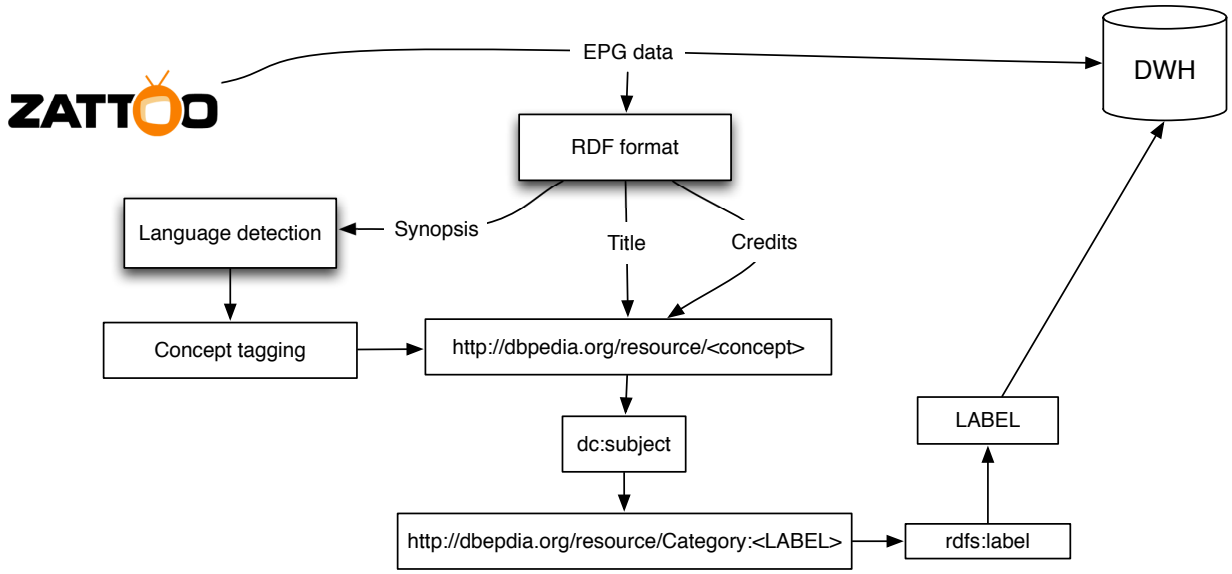


Figure 2: Zattoo workflow

[3] http://purl.org/ontology/po/
[4] http://lupedia.ontotext.com/
[5] http://notube.tv/
[6] http://www.alchemyapi.com/api/lang/

### 3.2.1 Zattoo RDF

In order to import Zattoo data into RDF format, we wrote a small script in Python. The input format looks like this:

```
<doc>
  <str name="id">9966901</str>
  <str name="title">Die allerbeste Sebastian Winkler Show</str>
  <str name="suggest_title">Die allerbeste Sebastian Winkler Show</str>
  <date name="start">2013-05-06T00:00:00Z</date>
  <str name="description">(Premiere in Einsfestival)</str>
  <str name="episode_title">mit Motsi Mabuse, Lady Bitch Ray und
       Sarah Brendel</str>
  <str name="image">54/f77c8d343e854467d4c6ba872715a7.jpg</str>
  <str name="dcid">100</str>
  <str name="grouping_title">
    Die allerbeste Sebastian Winkler Show mit Motsi Mabuse, Lady Bitch Ray
    und Sarah Brendel
  </str>
  <str name="suggest_group_title">
    Die allerbeste Sebastian Winkler Show mit Motsi Mabuse, Lady Bitch Ray
    und Sarah Brendel
  </str>
  <date name="end">2013-05-06T00:30:00Z</date>
  <str name="common_ch_name">einsfestival</str>
  <str name="dcname">einsfestival.AxelSpringer</str>
  <arr name="tag">
   <str>Show</str>
   <str>Portrat</str>
   <str>Kultur und Musik</str>
   <str>Unterhaltung</str>
   <str>Talk</str>
   <str>Kultur</str>
  </arr>
  <arr name="label">
   <str>Portrat</str>
   <str>Kultur und Musik</str>
   <str>Unterhaltung</str>
   <str>Talk</str>
   <str>Kultur</str>
  </arr>
  <arr name="newcategory">
   <str>Documentary</str>
  </arr>
  <arr name="role">
   <str>guest</str>
   <str>guest</str>
   <str>guest</str>
  </arr>
  <arr name="credit">
   <str>Motsi Mabuse</str>
   <str>Lady Bitch Ray</str>
   <str>Sarah Brendel</str>
  </arr>
</doc>
```

Listing 1: XML representation of a Zattoo program

And the output format looks like this:

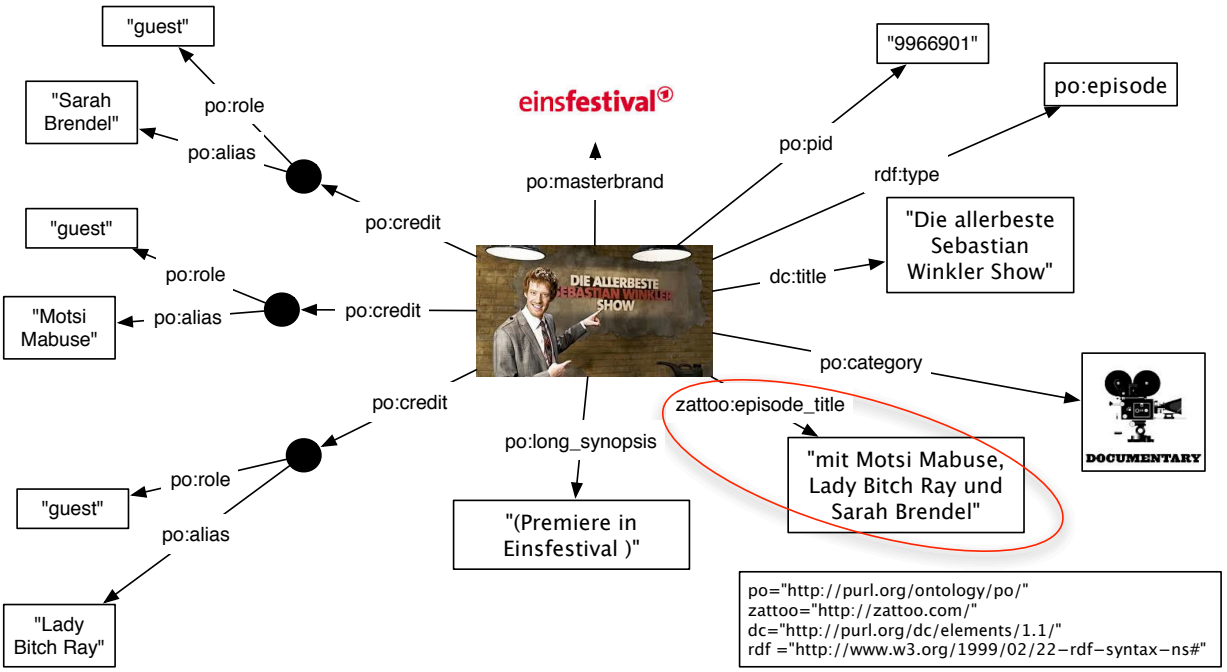


Figure 3: Zattoo RDF

As shown by Figure 3 (see Listing 4 in Appendix A for the written version), the Programmes Ontology did not fit perfectly with Zattoo EPG format. The ontology is modeled on the BBC EPG format, which describes the programs with their own title, regardless if it is an episode title or the main title. The main title of a program, in case it is an episode of a series, is provided in the version program description. Zattoo, instead, provide directly in the EPG description of every program both episode (*episode_title*) and main title (*title*). This aspect required an adaptation of the ontology: we added a new property *zattoo:episode_title*. Another aspect to notice is that Zattoo provide only one synopsis per program, while BBC provides small, medium and long synopses per program. We represent the Zattoo synopsis with the *po:long_synopsis* property. The other metadata is mapped easily into the properties.

# 4 Data Warehouse Population

Given the differences in the approach we adopted for the external data service, we use two different processes to populate the Data Warehouse: one for BBC and one for Zattoo. We provide EPG data and external data for the programs scheduled for the next seven days. Every night two scripts run in order to add the coming seventh day.

The population of the DWH is performed by means of two Python scripts, which make use of API developed by us to retrieve the external data. The workflow is as follows:

1. retrieve the EPG data in XML format and store it in the DWH
2. call the External Data API, which
   (a) retrieve the RDF description of the program
   (b) retrieve the external data for the current program metadata, namely: title, synopsis, credits
   (c) give a response in JSON format
3. read the response of the API and store it in the DWH

In the DWH the external data is described with a name, for instance *DBpedia category*, the label of the external data, for instance *1990s British television series*, and the URI of external data, for instance `http://dbpedia.org/resource/Category:1990s_British_television_series`. The external data is stored in dedicated tables in the DWH, one for the external data of the programs, *programme_annotations*, one for

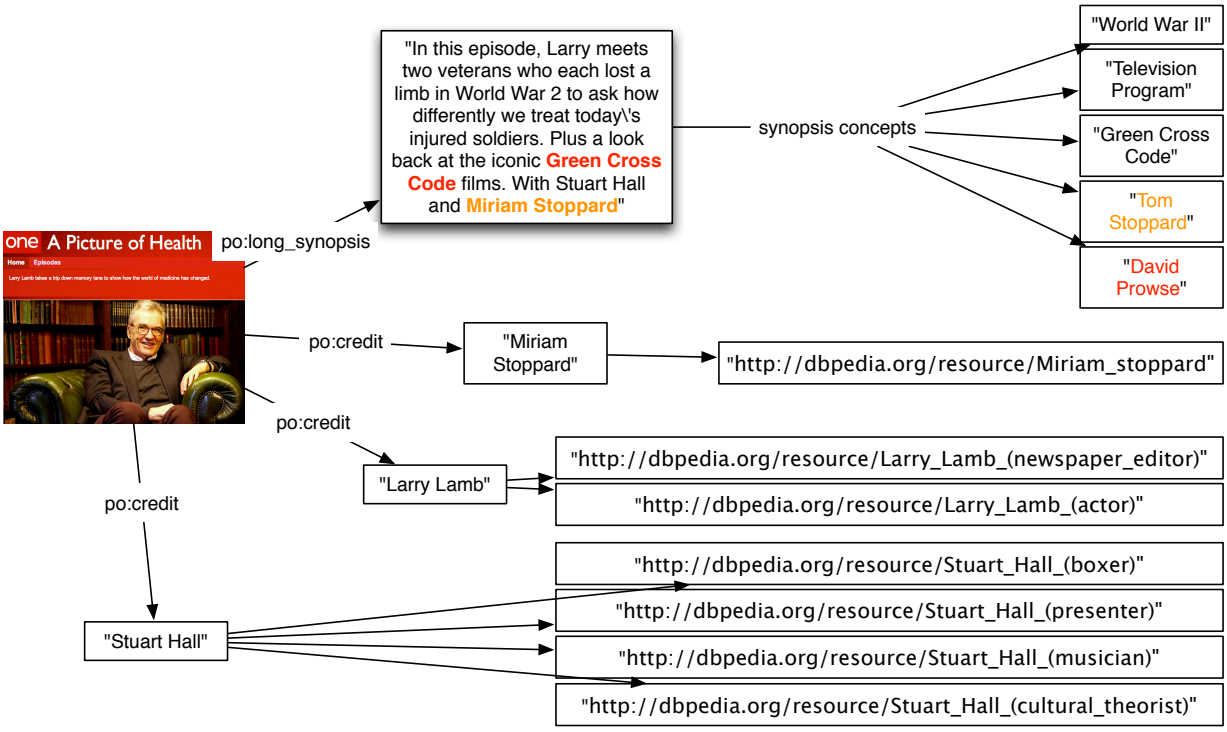


Figure 4: External Data API response.

the credits, *credits_annotations*, and a third one for the external data about the categories of the programmes, *categories_annotations*. The last table is designed to contain categories about the programmes coming from additional sources. Currently this table is empty. For a detailed description of the DWH schema, we refer the reader to Deliverable 3.2.

## 4.1 External Data API

The External Data API is available to all the partners which may need it. The current root is `http://eculture2.cs.vu.nl:1903/`. We developed two different services for the two broadcasters: *enrichBBCProgramme* and *enrichZattooProgramme*. The only input required is the program id. An example of call for a BBC program is: `http://eculture2.cs.vu.nl:1903/enrichBBCProgramme?pid=b01d0f6g&indent=true`. In Figure 4 we exemplify the response. For the complete JSON response, see Appendix B.

The colors in Figure 4 indicate why some concepts were tagged. In particular David Prowse was identified as he was involved in the Green Cross Code films, while Tom Stoppard was probably tagged due to the name Miriam Stoppard. Tag like this can generally be considered wrong, however in this case these two people are married, so in this particular case, the tag can be considered right.

This example is very peculiar also because it shows one of the issues we intend to address in the near future, the disambiguation of LOD entities. Indeed looking at Figure 4, we see that Stuart Hall is matched with more than one entity, namely: Stuart Hall (musician), Stuart Hall (presenter), Stuart Hall (cultural theorist), Stuart Hall (boxer). The disambiguation service will reduce this problem to a minimum, attaching a probability of rightness to every entity, to be able to select the right one. We will describe this service in detail in Section 6.

# 5 Data Warehouse API

Based on the requirements from TUDo and UZH we implemented a first version of the DWH API for ViSTA as a RESTful web service. The available functions and result formats are described below. The current web service root is `https://vista-tv.rapid-i.com/RAWS/process/`.

## 5.1 Functions

**metaDataByChannel**

- Retrieve current meta data for a show running on a specific channel. A show is currently running if start_time < now() AND end_time > now().
- Parameters:
  - CHANNEL_ID: ID of the channel, e.g. bbc_one
  - SOURCE: Source of the meta data, can be either BBC or Zattoo
- Example: Retrieve meta data for all shows that are currently running on BBC One
  `https://vista-tv.rapid-i.com/RAWS/process/metaDataByChannel?channel_id=bbc_one&source=bbc`

**metaDataByProgramme**

- Retrieve meta data for a specific show
- Parameters:
  - PROGRAMME_ID: The ID of the programme, e.g. b0074fx8
  - SOURCE: Source of the meta data, can be either BBC or Zattoo
- Example: Retrieve meta data for programme with programme ID b0074fx8 from BBC
  `https://vista-tv.rapid-i.com/RAWS/process/metaDataByProgramme?programme_id=b0074fx8&source=bbc`

**metaDataByChannelAndTime** Retrieve meta data for shows that started in a specified time period on a specific channel. The time period includes start and excludes end time.

- Parameters:
  - CHANNEL_ID: ID of the channel, e.g. bbc_one
  - START: Start time of the interval. The time format has to be yyyy-MM-dd HH:mm:ss, see example below that shows how to escape it.
  - END_TIME: End time of the interval. The time format has to be yyyy-MM-dd HH:mm:ss, see example below that shows how to escape it.
  - SOURCE: Source of the meta data, can be either BBC or Zattoo
- Example: Retrieve meta data for all shows that started between (2012-12-13 00:00:00,2012-12-13 01:00:00] on BBC Three
  `https://vista-tv.rapid-i.com/RAWS/process/metaDataByChannelAndTime?channel_id=bbc_three&start=2012-12-13+00%3A00%3A00&end=2012-12-13+01%3A00%3A00&source=bbc`

**categoryAnnotationByCategoryID** Retrieve a category annotation for a specific category ID

- Parameters:
  - CAT_ID: ID of the category
  - SOURCE: Source of the meta data, can be either BBC or Zattoo
- Example:
  `https://vista-tv.rapid-i.com/RAWS/process/categoryAnnotationByCategoryID?cat_id=C00001&source=bbc`

**creditAnnotationByCreditID** Retrieve a credit annotation for a specific credit ID

- Parameters:
  - CREDIT_ID: ID of the credit
  - SOURCE: Source of the meta data, can be either BBC or Zattoo
- Example:
  `https://vista-tv.rapid-i.com/RAWS/process/creditAnnotationByCreditID?credit_id=75&source=bbc`

## 5.2 Result formats

The results of *metaDataByChannel*, *metaDataByProgramme* and *metaDataByChannelAndTime* look like this:

Listing 2: Response example for *metaDataByChannel*, *metaDataByProgramme* and *metaDataByChannelAndTime*

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <programme>
    <pid>...</pid>
    <title>...</title>
    <short_synopsis>...</short_synopsis>
    <medium_synopsis>...</medium_synopsis>
    <long_synopsis>...</long_synopsis>
    <main_programme_pid>...</main_programme_pid>
      <start_time>...</start_time>
      <end_time>...</end_time>
      <channel_id>...</channel_id>
      <channel_name>...</channel_name>
      <channel_image>...</channel_image>
      <channel_common_name>...</channel_common_name>
      <annotations>
        <annotation>
          <annotation_name>...</annotation_name>
          <annotation_value>...</annotation_value>
          <annotation_URIs>...</annotation_URIs>
        </annotation>
      </annotations>
      <credits>
        <credit>
          <credit_id>...</credit_id>
          <role>...</role>
          <credit_name>...</credit_name>
          <annotations>
            <annotation>
              <annotation_name>...</annotation_name>
              <annotation_value>...</annotation_value>
              <annotation_URIs>...</annotation_URIs>
            </annotation>
          </annotations>
        </credit>
        <credit>
          <credit_id>...</credit_id>
          <role>...</role>
          <credit_name>...</credit_name>
          <annotations>
            <annotation>
              <annotation_name>...</annotation_name>
```

```
            <annotation_value>...</annotation_value>
            <annotation_URIs>...</annotation_URIs>
          </annotation>
        </annotations>
      </credit>
    </credits>
    <categories>
      <category>
        <cat_id>...</cat_id>
        <cat_type>...</cat_type>
        <cat_name>...</cat_name>
        <cat_broader>...</cat_broader>
        <annotations>
          <annotation>
            <annotation_name>...</annotation_name>
            <annotation_value>...</annotation_value>
            <annotation_URIs>...</annotation_URIs>
          </annotation>
        </annotations>
      </category>
    </categories>
  </programme>
</result>
```

The results of *categoryAnnotationByCategoryID* and *creditAnnotationByCreditID* look like this:

Listing 3: Response example for *categoryAnnotationByCategoryID* and *creditAnnotationByCreditID*

```
<?xml version="1.0" encoding="UTF-8"?>
<results serviceId="creditAnnotationByCreditID">
  <example-set>
    <example>
      <credit_id>75</credit_id>
      <annotation_name>DB Category</annotation_name>
      <annotation_value>British soap opera actors</annotation_value>
      <annotation_URIs>http://dbpedia.org/resource/Danielle_Harold
      </annotation_URIs>
    </example>
  </example-set>
</results>
```

### 5.3 Authentication

For authentication every partner has a login to access the Virtual Server/SQL Server and DWH API.

## 6 Future work

We plan to provide more services in the coming year. The main concern will be dedicated in providing more external data source, especially to be able to supply external evaluations of the programs, such reviews, from professionals and not, and ratings. The external data sources which are planned to be added next are IMDB and Rotten Tomatoes.
At the same time we really need a disambiguation approach in place. This is very important not only for people, as we saw in Section 4.1, but also for titles of programs. The idea under development is to use metadata, synopses and subtitles, when available, to create an information context, which will be used later to 'contextualize the entity extracted. For instance, in the previous example Stuart Hall has the role presenter in the program. This will help in deciding the entity to use, which is Stuart Hall (presenter). However, imagine the case where

the program at hand is a sport program: how could we have disambiguated between Stuart Hall (presenter) and Stuart Hall (boxer)?
Another service we want to provide is external data for user models. In particular we want to investigate the possibility of finding new connections between users and programs taking advantage of the LOD connectivity property.

# A Zattoo RDF

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
   xmlns:po="http://purl.org/ontology/po/"
   xmlns:zattoo="http://zattoo.com/"
   xmlns:dc="http://purl.org/dc/elements/1.1/"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">

 <rdf:Description rdf:about="http://zattoo.com/programme/9966901">
  <po:pid>9966901</po:pid>
  <po:masterbrand rdf:resource="http://zattoo.com/ einsfestival "/>
  <zattoo:episode_title>mit Motsi Mabuse, Lady Bitch Ray und Sarah Brendel</zattoo:episode_title>
  <po:long_synopsis>(Premiere in Einsfestival )</po:long_synopsis>
  <po:credit rdf:nodeID="N4a5f3095a3b34502b04e110336e2f6ac"/>
  <po:credit rdf:nodeID="N860dfe7af1ad4df78e795587d23aad25"/>
  <po:credit rdf:nodeID="N904c763f46cc46448c949cf010ea3ed9"/>
  <po:category rdf:resource=zattoo:Documentary/>
  <rdf:type rdf:resource=po:episode/>
  <dc:title>Die allerbeste Sebastian Winkler Show</dc:title>
 </rdf:Description>
 <rdf:Description rdf:nodeID="N904c763f46cc46448c949cf010ea3ed9">
  <rdf:type rdf:resource=po:credit/>
   <po:role>guest</po:role>
   <po:alias>Sarah Brendel</po:alias>
 </rdf:Description>
 <rdf:Description rdf:nodeID="N4a5f3095a3b34502b04e110336e2f6ac">
  <rdf:type rdf:resource=po:credit/>
   <po:role>guest</po:role>
   <po:alias>Motsi Mabuse</po:alias>
 </rdf:Description>
 <rdf:Description rdf:nodeID="N860dfe7af1ad4df78e795587d23aad25">
  <rdf:type rdf:resource=po:credit/>
   <po:role>guest</po:role>
   <po:alias>Lady Bitch Ray</po:alias>
 </rdf:Description>
</rdf:RDF>
```

Listing 4: RDF representation of a Zattoo program

# B Zattoo RDF

```
[
 {
  "pid":"b01d0f6g",
  "synopsis_enrichments_concepts": [
      {
  "category":"World␣War␣II",
  "category_URIs": [
    "http://dbpedia.org/resource/World_War_II"
  ]
      },
      {
  "category":" Television ␣program",
  "category_URIs": [
    "http://dbpedia.org/resource/Television_program"
```

```
    ]
        },
        {
    "category":"Green␣Cross␣Code",
    "category_URIs": [
      "http://dbpedia.org/resource/Green_Cross_Code"
    ]
        },
        {
    "category":"Tom␣Stoppard",
    "category_URIs": [
      "http://dbpedia.org/resource/Tom_Stoppard"
    ]
        },
        {
    "category":"David␣Prowse",
    "category_URIs": [
      "http://dbpedia.org/resource/David_Prowse"
    ]
        }
      ],
    "credits": [
     {
    "name":"Larry␣Lamb",
    "nameURIs":"http://dbpedia.org/resource/Larry_Lamb_(actor)",
    "categories": [
      "1947␣births",
      "People␣from␣Edmonton,␣London",
      "English␣film␣actors",
      "English␣television␣actors",
      "Living␣people",
      "Actors␣from␣London",
      "English␣Christians"
    ],
    "uris": [
     "http://dbpedia.org/resource/Category:1947_births",
     "http://dbpedia.org/resource/Category:People_from_Edmonton,_London",
     "http://dbpedia.org/resource/Category:English_film_actors",
     "http://dbpedia.org/resource/Category:English_television_actors",
     "http://dbpedia.org/resource/Category:Living_people",
     "http://dbpedia.org/resource/Category:Actors_from_London",
     "http://dbpedia.org/resource/Category:English_Christians"
    ]
   },
   {
    "name":"Larry␣Lamb",
    "nameURIs":"http://dbpedia.org/resource/Larry_Lamb_(newspaper_editor)",
    "categories": [
     "1929␣births",
     "2000␣deaths",
     "British␣journalists",
     "British␣newspaper␣editors",
     "Knights␣Bachelor",
     "Daily␣Express␣people"
    ],
    "uris": [
     "http://dbpedia.org/resource/Category:1929_births",
     "http://dbpedia.org/resource/Category:2000_deaths",
```

```
    "http://dbpedia.org/resource/ Category:British_journalists ",
    "http://dbpedia.org/resource/Category:British_newspaper_editors",
    "http://dbpedia.org/resource/Category:Knights_Bachelor",
    "http://dbpedia.org/resource/Category:Daily_Express_people"
   ]
  },
  {
   "name":"Miriam␣Stoppard",
   "nameURIs":"http://dbpedia.org/resource/Miriam_Stoppard",
   " categories ": [
    "1937␣births",
    " British ␣advice␣columnists",
    "English␣Jews",
    "English␣ television ␣presenters ",
    "20th−century␣English␣medical␣doctors",
    "Alumni␣of␣Durham␣University",
    "Alumni␣of␣Newcastle␣University",
    "Living␣people",
    "People␣from␣Newcastle␣upon␣Tyne",
    " Officers ␣of␣the␣Order␣of␣the␣British␣Empire",
    "People␣educated␣at␣Central␣Newcastle␣High␣School"
   ],
   " uris ": [
    "http://dbpedia.org/resource/Category:1937_births",
    "http://dbpedia.org/resource/Category:British_advice_columnists",
    "http://dbpedia.org/resource/Category:English_Jews",
    "http://dbpedia.org/resource/ Category:English_television_presenters ",
    "http://dbpedia.org/resource/Category:20th−century_English_medical_doctors",
    "http://dbpedia.org/resource/Category:Alumni_of_Durham_University",
    "http://dbpedia.org/resource/Category:Alumni_of_Newcastle_University",
    "http://dbpedia.org/resource/Category:Living_people",
    "http://dbpedia.org/resource/Category:People_from_Newcastle_upon_Tyne",
    "http://dbpedia.org/resource/Category:Officers_of_the_Order_of_the_British_Empire",
    "http://dbpedia.org/resource/Category:People_educated_at_Central_Newcastle_High_School"
    ]
   },
   {
    "name":"Stuart␣Hall",
    "nameURIs":"http://dbpedia.org/resource/Stuart_Hall_(boxer)",
    " categories ": [
   "1980␣births",
   "Living␣people",
   "English␣boxers",
   "Bantamweight␣boxers"
    ],
    " uris ": [
   "http://dbpedia.org/resource/Category:1980_births",
   "http://dbpedia.org/resource/Category:Living_people",
   "http://dbpedia.org/resource/Category:English_boxers",
   "http://dbpedia.org/resource/Category:Bantamweight_boxers"
    ]
   },
   {
    "name":"Stuart␣Hall",
    "nameURIs":"http://dbpedia.org/resource/Stuart_Hall_(cultural_theorist)",
    " categories ": [
   " British ␣ literary ␣ critics ",
   "Jamaican␣Rhodes␣scholars",
```

```
"Alumni␣of␣Merton␣College,␣Oxford",
" British ␣ sociologists ",
" British ␣academics",
"1932␣births",
"Living␣people",
"Academics␣of␣the␣Open␣University",
"Academics␣of␣the␣University␣of␣Birmingham",
"English␣people␣of␣Jamaican␣descent",
"Black␣ British ␣ writers ",
"Marxist␣humanists"
],
" uris ": [
  "http://dbpedia.org/resource/ Category:British_literary_critics ",
  "http://dbpedia.org/resource/Category:Jamaican_Rhodes_scholars",
  "http://dbpedia.org/resource/Category:Alumni_of_Merton_College,_Oxford",
  "http://dbpedia.org/resource/ Category:British_sociologists ",
  "http://dbpedia.org/resource/Category:British_academics",
  "http://dbpedia.org/resource/Category:1932_births",
  "http://dbpedia.org/resource/Category:Living_people",
  "http://dbpedia.org/resource/Category:Academics_of_the_Open_University",
  "http://dbpedia.org/resource/Category:Academics_of_the_University_of_Birmingham",
  "http://dbpedia.org/resource/Category:English_people_of_Jamaican_descent",
  "http://dbpedia.org/resource/Category:Black_British_writers",
  "http://dbpedia.org/resource/Category:Marxist_humanists"
]
    },
    {
"name":"Stuart␣Hall",
"nameURIs":"http://dbpedia.org/resource/Stuart_Hall_(musician)",
" categories ": [
  "Academics␣of␣Middlesex␣University",
  "Living␣people",
  "Year␣of␣birth␣missing␣( living ␣people)"
],
" uris ": [
  "http://dbpedia.org/resource/Category:Academics_of_Middlesex_University",
  "http://dbpedia.org/resource/Category:Living_people",
  "http://dbpedia.org/resource/Category:Year_of_birth_missing_(living_people)"
]
    },
    {
"name":"Stuart␣Hall",
"nameURIs":"http://dbpedia.org/resource/Stuart_Hall_(presenter)",
" categories ": [
  "Alumni␣of␣the␣University␣of␣Manchester␣Institute␣of␣Science␣and␣Technology",
  " British ␣sports␣broadcasters",
  " British ␣ association ␣ football ␣commentators",
  "English␣ journalists ",
  "English␣radio␣ personalities ",
  "English␣ television ␣ presenters ",
  "People␣from␣Hyde,␣Greater␣Manchester",
  "1929␣births",
  "Living␣people",
  " Officers ␣of␣the␣Order␣of␣the␣ British ␣Empire"
],
" uris ": [
  "http://dbpedia.org/resource/
   Category:Alumni_of_the_University_of_Manchester_Institute_of_Science_and_Technology",
```

```
      "http://dbpedia.org/resource/Category:British_sports_broadcasters",
      "http://dbpedia.org/resource/Category:British_association_football_commentators",
      "http://dbpedia.org/resource/ Category:English_journalists ",
      "http://dbpedia.org/resource/ Category:English_radio_personalities ",
      "http://dbpedia.org/resource/ Category:English_television_presenters ",
      "http://dbpedia.org/resource/Category:People_from_Hyde,_Greater_Manchester",
      "http://dbpedia.org/resource/Category:1929_births",
      "http://dbpedia.org/resource/Category:Living_people",
      "http://dbpedia.org/resource/Category:Officers_of_the_Order_of_the_British_Empire"
    ]
        }
      ],
      "DBpedia categories": []
    }
  ]
```

Listing 5: External Data API Response

# C Programmes Ontology

This appendix contains the specification of the Programmes Ontology (source: `http://www.bbc.co.uk/ontologies/programmes/2009-09-07.n3`).

```
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix vs: <http://www.w3.org/2003/06/sw-vocab-status/ns#>.
@prefix mo: <http://purl.org/ontology/mo/>.
@prefix tl: <http://purl.org/NET/c4dm/timeline.owl#>.
@prefix event: <http://purl.org/NET/c4dm/event.owl#>.
@prefix dc: <http://purl.org/dc/elements/1.1/>.
@prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>.
@prefix tags: <http://www.holygoat.co.uk/owl/redwood/0.1/tags/>.
@prefix skos: <http://www.w3.org/2004/02/skos/core#>.
@prefix dcterms: <http://purl.org/dc/terms/>.
@prefix frbr: <http://purl.org/vocab/frbr/core#>.
@prefix : <http://purl.org/ontology/po/>.

<http://purl.org/ontology/po/>
  a owl:Ontology;
  rdfs:label "Programmes ontology";
  rdfs:comment """
    A vocabulary for programme data.
    It defines concepts such as brands, series, episodes, broadcasts, etc.
  """;
  dc:creator <http://moustaki.org/foaf.rdf#moustaki>;
  dc:contributor <http://www.aelius.com/njh#me>;
  dc:contributor <http://metade.org/foaf.rdf#me>;
  dc:date "$Date: 2009/02/20 16:00:00 $";
    owl:imports
        <http://purl.org/NET/c4dm/event.owl>,
        <http://purl.org/NET/c4dm/timeline.owl>
  .
```

```
# I'll use this to capture the status of the terms
# defined in this ontology
vs:term_status a owl:AnnotationProperty.


# Concepts

# Content

:Programme
  a owl:Class;
  rdfs:label "programme";
  rdfs:comment """
    A programme, can either be a brand, a series or an episode
  """;
  vs:term_status "stable";
  .

:Brand
  a owl:Class;
  rdfs:label "brand";
  rdfs:comment """
    A brand, e.g. 'Top Gear'
  """;
  rdfs:subClassOf :Programme;
  owl:disjointWith :Episode;
    owl:disjointWith :Clip;
  owl:disjointWith :Series;
  vs:term_status "stable";
  .

:ProgrammeItem
    a owl:Class;
    rdfs:label "programme item";
    rdfs:comment """
        A programme that can have versions, and as such can be broadcast or made available on-
    demand, e.g. a clip or an episode.
    """;
    rdfs:subClassOf :Programme;
    owl:disjointWith :Brand;
    owl:disjointWith :Series;
    vs:term_status "testing";
    .

:Episode
  a owl:Class;
  rdfs:label "episode";
  rdfs:comment """
    A particular episode, e.g. 'Top Gear, first episode of the first series' or the film 'A Walk
    in the Sun' (http://www.bbc.co.uk/programmes/b00gfzdt)
  """;
  rdfs:subClassOf :Programme;
    rdfs:subClassOf :ProgrammeItem;
  owl:disjointWith :Series;
    owl:disjointWith :Brand;
  vs:term_status "stable";
  .
```

```
:Clip
    a owl:Class;
    rdfs:label "clip";
    rdfs:comment """
        A particular clip, e.g. 'Clip of Top Gear, first series'
    """;
    rdfs:subClassOf :Programme;
    rdfs:subClassOf :ProgrammeItem;
    owl:disjointWith :Series;
    owl:disjointWith :Brand;
    vs:term_status "testing";
    .

:Series
  a owl:Class;
  rdfs:label "series";
  rdfs:comment """
    A series, e.g. 'Top Gear, first season'
  """;
  rdfs:subClassOf :Programme;
  vs:term_status "stable";
  .

:Category
    a owl:Class;
    # Not consistent with the SKOS reference -- we'll leave it aside for now
    # rdfs:subClassOf skos:Concept;
    rdfs:label "category";
    rdfs:comment """
        A category provides a way of classifying a set of programmes. Such classifications
        can be performed according to multiple dimensions and taxonomies, e.g. genre, format,
     places, people, subjects ...
    """;
    vs:term_status "testing";
    .

:Genre
    a owl:Class;
    rdfs:subClassOf :Category;
  rdfs:label "genre";
  rdfs:comment """
        An anchor point for a programmes' genre taxonomy, e.g. 'Drama'/'Biographical'.
  """;
  vs:term_status "testing";
  .

:Format
    a owl:Class;
    rdfs:subClassOf :Category;
    rdfs:label "format";
    rdfs:comment """
        Anchor point for format taxonomies, similar to po:Genre for genre taxonomies.
        Instances of this concept include documentaries, talk shows, animation, etc.
    """;
    vs:term_status "testing";
    .
```

```
:Subject
    a owl:Class;
    rdfs:subClassOf :Category;
    rdfs:label "subject";
    rdfs:comment """
        Anchor point for subject taxonomies.
    """;
    vs:term_status "testing";
    .

:Place
    owl:equivalentClass geo:Feature;
    a owl:Class;
    rdfs:label "place";
    rdfs:comment "A physical place";
    rdfs:subClassOf :Category;
    vs:term_status "testing";
    .
:Person
    owl:equivalentClass foaf:Person;
    a owl:Class;
    rdfs:label "person";
    rdfs:comment "A person";
    rdfs:subClassOf :Category;
    vs:term_status "testing";
    .

# Publishing

:Version
  a owl:Class;
  rdfs:label "version";
  rdfs:comment """
    A particular version of an episode.
    Such versions include shortened ones, audio described ones
    or ones that holds sign language.
    The version is associated to a timeline.
  """;
  vs:term_status "stable";
  .

:OriginalVersion
  a owl:Class;
  rdfs:label "original version";
  rdfs:comment """
    An 'original' version, the legacy version of
    a particular episode.
  """;
  rdfs:subClassOf :Version;
    owl:disjointWith :ShortenedVersion;
  vs:term_status "testing";
  .
:AudioDescribedVersion
  a owl:Class;
  rdfs:label "audio description";
  rdfs:comment """
    A version holding an audio description.
  """;
```

```
    rdfs:subClassOf :Version;
    vs:term_status "testing";
    .
:SignedVersion
    a owl:Class;
    rdfs:label "sign language";
    rdfs:comment """
      A version holding sign language.
    """;
    rdfs:subClassOf :Version;
    vs:term_status "testing";
    .
:ShortenedVersion
    a owl:Class;
    rdfs:label "shortened version";
    rdfs:comment """
      A shortened version.
    """;
    rdfs:subClassOf :Version;
    owl:disjointWith :OriginalVersion;
    vs:term_status "testing";
    .

:Broadcast
    a owl:Class;
    rdfs:label "broadcast";
    rdfs:comment """
      A broadcast event.
      Subsumes the event concept defined in
      the Event ontology.
      A broadcast is associated with a service, and with a
       particular version of an episode.
    """;
    rdfs:subClassOf event:Event;
    vs:term_status "stable";
    .
:FirstBroadcast
    a owl:Class;
    rdfs:label "first broadcast";
    rdfs:comment """
      Specifies a broadcast as being the
      first one of a particular version.
    """;
    rdfs:subClassOf :Broadcast;
    owl:disjointWith :RepeatBroadcast;
    vs:term_status "testing";
    .
:RepeatBroadcast
    a owl:Class;
    rdfs:label "repeat";
    rdfs:comment """
      Specifies a broadcast as being a
      repeat.
    """;
    rdfs:subClassOf :Broadcast;
    owl:disjointWith :FirstBroadcast;
    vs:term_status "testing";
    .
```

```
:Season
  a owl:Class;
  rdfs:label "season";
  rdfs:comment """
    A season is a group of broadcasts.
  """;
  vs:term_status "testing";
  .

:Broadcaster
  a owl:Class;
  rdfs:label "broadcaster";
  rdfs:subClassOf foaf:Organization;
  rdfs:comment """
    An organization responsible of some broadcasting services.
        It can hold a set of services and outlets.
  """;
  vs:term_status "stable";
  .

:Service
  a owl:Class;
  rdfs:label "service";
  rdfs:comment """
        A broadcasting service.
    Instances of this concept include BBC Radio Wales, BBC Radio 4, BBC News, etc.
        A service is a collection of outlets which contain common material, but with some
    variations, e.g. by
        region.
        Hence, a service may have multiple outlets (po:Outlet), e.g. BBC Radio 4 has BBC Radio 4
    LW and BBC Radio 4 FM.
        A hierarchy of services types is defined within this ontology, e.g. radio and TV.
        A service that is a master brand only (a service that only commissions programmes, e.g.
    BBC Switch) should
        be an instance of the top-level po:Service.
  """;
  vs:term_status "stable";
  .

:Outlet
    a owl:Class;
    rdfs:subClassOf :Service;
    rdfs:label "outlet";
    rdfs:comment """
        Outlet of a particular service, e.g. Radio 4 LW and FM for Radio 4.
        Outlets are services which do not have variations.
        The identity criteria for an outlet is its timeline.
        For example, Radio 4 LW broadcasts on Analogue Long Wave, but also on Digital Satellite.
        It corresponds to just one outlet, as they are simulcasts.
        The two physical channels for broadcasts correspond to po:Channel.
    """;
    vs:term_status "testing";
    .

:Channel
    a owl:Class;
    rdfs:label "physical channel";
```

```
      rdfs:comment """
          A physical channel on which a broadcast occurs.
          A single outlet or service can be associated with multiple channels.
          For example, Radio 4 LW broadcasts on Analogue Long Wave and on Digital
           Satellite .
      """;
      vs:term_status "testing";
      .

# Hierarchy of services

:Radio
  a owl:Class;
  rdfs:label "radio";
  rdfs:comment """
    Services that use a radio medium.
  """;
  rdfs:subClassOf :Service;
  vs:term_status "testing";
  owl:disjointWith :TV;
  owl:disjointWith :Web;
  .
:LocalRadio
  a owl:Class;
  rdfs:label "radio";
  rdfs:comment """
    Radio services aiming at a local coverage.
  """;
  rdfs:subClassOf :Radio;
  vs:term_status "testing";
  owl:disjointWith :RegionalRadio;
  owl:disjointWith :NationalRadio;
  .
:RegionalRadio
  a owl:Class;
  rdfs:label "regional radio";
  rdfs:comment """
    Radio services aiming at a regional coverage.
  """;
  rdfs:subClassOf :Radio;
  vs:term_status "testing";
  owl:disjointWith :NationalRadio;
  .
:NationalRadio
  a owl:Class;
  rdfs:label "national radio";
  rdfs:comment """
    Radio services aiming at a national coverage.
  """;
  rdfs:subClassOf :Radio;
  vs:term_status "testing";
  .
:TV
  a owl:Class;
  rdfs:label "tv";
  rdfs:comment """
    Services that use a television medium.
  """;
```

```
    rdfs:subClassOf :Service;
    vs:term_status "testing";
    owl:disjointWith :Web;
    .
:Web
    a owl:Class;
    rdfs:label "web";
    rdfs:comment """
      Services that use a Web medium.
    """;
    rdfs:subClassOf :Service;
    vs:term_status "testing";
    .

# Some "physical" channels

:FM
    a owl:Class;
    rdfs:subClassOf :Channel;
    rdfs:label "FM";
    rdfs:comment """
      The FM broadcast band
    """;
    vs:term_status "testing";
    .
:LW
    a owl:Class;
    rdfs:subClassOf :Channel;
    rdfs:label "AM";
    rdfs:comment """
      The AM broadcast band
    """;
    vs:term_status "testing";
    .
:DAB
    a owl:Class;
    rdfs:subClassOf :Channel;
    rdfs:label "DAB";
    rdfs:comment """
      Digital Audio Broadcasting
    """;
    vs:term_status "testing";
    .
:DVB
    a owl:Class;
    rdfs:subClassOf :Channel;
    rdfs:label "DAB";
    rdfs:comment """
      Digital Video Broadcasting
    """;
    vs:term_status "testing";
    .
:IPStream
    # owl:equivalentClass mo:Stream ?
    a owl:Class;
    rdfs:subClassOf :Channel;
    rdfs:label "IP stream";
    rdfs:comment """
```

```
    IP stream
  """;
  vs:term_status "testing";
  .

# Segmentation

:Segment
  a owl:Class;
  rdfs:subClassOf event:Event;
  rdfs:label "segment";
  rdfs:comment """
    Classification of an episode version's region, e.g. 'this track was played at that time'.
  """;
  vs:term_status "testing";
  .
:MusicSegment
  a owl:Class;
  rdfs:subClassOf :Segment;
  rdfs:comment """
    Classification of an episode version's region corresponding to a musical track being played.
  """;
  vs:term_status "testing";
  .
:SpeechSegment
    a owl:Class;
    rdfs:subClassOf :Segment;
    rdfs:comment """
        Classification of an episode version's region holding speech content.
    """;
    vs:term_status "testing";
    .
: Subtitle
    a owl:Class;
    rdfs:subClassOf event:Event;
    rdfs:label "subtitle";
    rdfs:comment """
        Classification of an episode version's region corresponding to a subtitle being shown.
    """;
    vs:term_status "testing";
    .

# Properties

# Object properties

:masterbrand
    a owl:ObjectProperty;
    rdfs:label "masterbrand";
    rdfs:comment "Associates a programme with its masterbrand (its commissionner)";
    rdfs:domain :Programme;
    rdfs:range :Service;
    vs:term_status "stable";
    .

:track
    a owl:ObjectProperty;
    rdfs:subPropertyOf event:factor;
```

```
    rdfs:label "track";
    rdfs:comment "Associates a music segment with a track, as defined in MO";
    rdfs:domain :MusicSegment;
    rdfs:range mo:Track;
    vs:term_status "testing";
    .

:season_broadcast
  a owl:ObjectProperty;
  rdfs:domain :Season;
  rdfs:range :Broadcast;
  rdfs:label "broadcast";
  rdfs:comment "Associates a season to its constituent broadcasts";
  vs:term_status "testing";
  .

:outlet
  a owl:ObjectProperty;
  rdfs:domain :Service;
  rdfs:range :Outlet;
  rdfs:label "outlet";
  rdfs:comment """
    Associates a service to an outlet, e.g. Radio 4 to Radio 4 LW.
  """;
  vs:term_status "testing";
  .

:channel
    a owl:ObjectProperty;
    rdfs:domain :Service;
    rdfs:range :Channel;
    rdfs:label "channel";
    rdfs:comment """
        Associates a service to a channel, e.g. Radio 4 LW to Radio 4 LW on Analogue Long Wave.
    """;
    vs:term_status "testing";
    .

:broadcaster
  a owl:ObjectProperty;
  rdfs:domain :Service;
  rdfs:range :Broadcaster;
  rdfs:label "broadcaster";
  rdfs:comment """
    Associates a service to a broadcaster
  """;
  vs:term_status "testing";
  .

:location
  a owl:ObjectProperty;
  rdfs:label "location";
  rdfs:subPropertyOf foaf:based_near;
  rdfs:comment """
    Associates a service to a geographic location,
    aiming at capturing what this service aims at covering.
  """;
  rdfs:domain :Service;
```

```
    rdfs:range geo:SpatialThing;
    vs:term_status "testing";
    .

:episode
    a owl:ObjectProperty;
    rdfs:label "episode";
    rdfs:comment """
        Associates a brand or a series to an episode constituting it.
    """;
    rdfs:domain [
        a owl:Class;
        owl:unionOf (:Brand :Series);
    ];
    rdfs:subPropertyOf dcterms:hasPart;
    rdfs:range :Episode;
    vs:term_status "testing";
    .

:clip
    a owl:ObjectProperty;
    rdfs:label "clip";
    rdfs:comment """
        Associates a brand, a series or an episode to a clip.
    """;
    rdfs:domain [
        a owl:Class;
        owl:unionOf (:Brand :Series :Episode);
    ];
    rdfs:subPropertyOf dcterms:hasPart;
    rdfs:range :Clip;
    vs:term_status "testing";
    .

:series
    a owl:ObjectProperty;
    a owl:TransitiveProperty;
    rdfs:label "series";
    rdfs:comment """
        Associates a brand or a series to a series constituting it.
    """;
    rdfs:domain [
        a owl:Class;
        owl:unionOf (:Brand :Series);
    ];
    rdfs:subPropertyOf dcterms:hasPart;
    rdfs:range :Series;
    vs:term_status "testing";
    .

:parent_series
    a owl:ObjectProperty;
    a owl:TransitiveProperty;
    rdfs:subPropertyOf dcterms:isPartOf;
    rdfs:label "parent serie";
    rdfs:comment """
        Relates a series to a series constituting it (eg. 'Waking the dead').
    """;
```

```
    rdfs:domain :Series;
    rdfs:range  :Series;
    vs:term_status "testing";
    .

:parent_service
    a owl:ObjectProperty;
    a owl: TransitiveProperty ;
    rdfs:subPropertyOf dcterms:isPartOf;
    rdfs: label  "parent  service ";
    rdfs:comment """
      Relates a service to another service encapsulating it (eg. 'BBC One' and 'BBC One South')
    """;
    rdfs:domain :Service;
    rdfs:range  :Service;
    vs:term_status "testing";
    .

:service  #master brand?
    a owl:ObjectProperty;
    a owl:FunctionalProperty;
    rdfs: label  " service ";
    rdfs:comment """
      Associate a brand, series or episode to the master brand service.
    """;
    rdfs:domain :Programme;
    rdfs:range  :Service;
    vs:term_status "testing";
    .

:version
    a owl:ObjectProperty;
    a owl: InverseFunctionalProperty ;
    rdfs: label  "version ";
    rdfs:comment """
      Associate an episode to a version of it.
      Different versions of a same episode can exist (shortened version, version
      designed for the hearing impaired, etc.).
    """;
    rdfs:domain :ProgrammeItem;
    rdfs:range  :Version;
    vs:term_status "testing";
    .

:broadcast_on
    a owl:ObjectProperty;
      owl: equivalentProperty  :broadcasted_on;
    rdfs:subPropertyOf event: factor;
    rdfs: label  "broadcast on";
    rdfs:comment """
      Relates a particular broadcast to the service or outlet on which it was on.
      Sub-property of the event: factor one.
    """;
    rdfs:domain :Broadcast;
    rdfs:range  :Service;
    vs:term_status "testing";
    .
```

```
:broadcasted_on
    a owl:ObjectProperty;
    rdfs:comment "Deprecated property, left here for historical reasons";
    .

:broadcast_of
  a owl:ObjectProperty;
  rdfs:subPropertyOf event:factor;
  rdfs:label "broadcast of";
  rdfs:comment """
    Relates a particular broadcast to the version being broadcasted.
    Sub-property of the event:factor one.
  """;
  rdfs:domain :Broadcast;
  rdfs:range :Version;
  vs:term_status "testing";
  .

:category
    a owl:ObjectProperty;
    rdfs:label "category";
    rdfs:comment """
        Relates a programme to a particular category, e.g. genre, format, place ...
    """;
    rdfs:domain :Programme;
    rdfs:range :Genre;
    vs:term_status "testing";
    .

:genre
  a owl:ObjectProperty;
    rdfs:subPropertyOf :category;
  rdfs:label "genre";
  rdfs:comment """
    Relates a programme to a particular genre.
  """;
  rdfs:domain :Programme;
  rdfs:range :Genre;
  vs:term_status "testing";
  .

:format
  a owl:ObjectProperty;
    rdfs:subPropertyOf :category;
  rdfs:label "format";
  rdfs:comment """
    Relates a programme to a particular format (eg. 'Animation', 'Documentary', etc.).
  """;
  rdfs:domain :Programme;
  rdfs:range :Format;
  vs:term_status "testing";
  .

:subject
    a owl:ObjectProperty;
    rdfs:subPropertyOf :category;
    rdfs:label "subject";
    rdfs:comment """
```

```
        Relates a programme to a subject (e.g. 'easter')
    """;
    rdfs:domain :Programme;
    rdfs:range :Subject;
    vs:term_status "testing";
    .

:place
    a owl:ObjectProperty;
    rdfs:subPropertyOf :category;
    rdfs:label "place";
    rdfs:comment """
        Relates a programme to a place (e.g. 'London')
    """;
    rdfs:domain :Programme;
    rdfs:range :Place;
    vs:term_status "testing";
    .

:person
    a owl:ObjectProperty;
    rdfs:subPropertyOf :category;
    rdfs:label "person";
    rdfs:comment """
        Relates a programme to a person
    """;
    rdfs:domain :Programme;
    rdfs:range :Person;
    vs:term_status "testing";
    .

# Credits

:credit
  a owl:ObjectProperty;
    rdfs:subPropertyOf dc:contributor ;
  rdfs:label "credit";
  rdfs:comment "Relates a programmes to a person who is credited in it";
  rdfs:domain :Programme;
  rdfs:range foaf:Agent;
  vs:term_status "testing";
  .
:actor
  a owl:ObjectProperty;
  rdfs:label "actor";
  rdfs:comment "Relates a programmes to one of its actors – a person who plays the role of a
    character"; # FIXME
  rdfs:domain :Programme;
  rdfs:range foaf:Agent;
  rdfs:subPropertyOf :credit ;
  vs:term_status "testing";
  .
:anchor
  a owl:ObjectProperty;
  rdfs:label "anchor";
  rdfs:comment "A television reporter who coordinates a programme";
  rdfs:domain :Programme;
  rdfs:range foaf:Agent;
```

```
    rdfs:subPropertyOf :credit ;
    vs:term_status "testing";
    .
: participant
    a owl:ObjectProperty;
    rdfs:label "participant";
    rdfs:comment "Relates a programme to one of its participants";
    rdfs:domain :Programme;
    rdfs:range foaf:Agent;
    rdfs:subPropertyOf :credit ;
    vs:term_status "testing";
    .
:commentator
    a owl:ObjectProperty;
    rdfs:label "commentator";
    rdfs:comment "Relates a programme to one of its commentators";
    rdfs:domain :Programme;
    rdfs:range foaf:Agent;
    rdfs:subPropertyOf :credit ;
    vs:term_status "testing";
    .
# key talent?
:executive_producer
    a owl:ObjectProperty;
    rdfs:label "executive producer";
    rdfs:comment "Relates a programme to its executive producer - a producer who is not involved in
      any technical aspects of the making process, but who is still responsible for the overall
      production. Typically an executive producer handles business and legal issues";
    rdfs:domain :Programme;
    rdfs:range foaf:Agent;
    rdfs:subPropertyOf :credit ;
    vs:term_status "testing";
    .
:performer
    a owl:ObjectProperty;
    rdfs:label "performer";
    rdfs:comment "Relates a programme to an entertainer who performs a dramatic or musical work for
      audience";
    rdfs:domain :Programme;
    rdfs:range foaf:Agent;
    rdfs:subPropertyOf :credit ;
    vs:term_status "testing";
    .
: director
    a owl:ObjectProperty;
    rdfs:label "director";
    rdfs:comment "Relates a programme to its supervisor. Generally refers to the person responsible
      for all audience-visible components of a program, film, or show, whereas the producer is
      responsible for the financial and other behind-the-scenes aspects. A director's duties might
      also include casting, script editing, shot selection, shot composition, and editing";
    rdfs:domain :Programme;
    rdfs:range foaf:Agent;
    rdfs:subPropertyOf :credit ;
    vs:term_status "testing";
    .
:author
    a owl:ObjectProperty;
    rdfs:label "author";
```

```
    rdfs:comment "Relates a programme to its author – the person who created the content";
    rdfs:domain :Programme;
    rdfs:range  foaf:Agent;
    rdfs:subPropertyOf :credit;
    vs:term_status "testing";
    .
:producer
    a owl:ObjectProperty;
    rdfs:label  "producer";
    rdfs:comment "Relates a programme to its producer – the manager of an event, show, or other work,
        usually the individual in charge of finance, personnel, and other non-artistic aspects in the
        development of commercials, plays, movies, and other works";
    rdfs:domain :Programme;
    rdfs:range  foaf:Agent;
    rdfs:subPropertyOf :credit;
    vs:term_status "testing";
    .
:news_reader
    a owl:ObjectProperty;
    rdfs:label  "news reader";
    rdfs:comment "Relates a programme to its news reader";
    rdfs:domain :Programme;
    rdfs:range  foaf:Agent;
    rdfs:subPropertyOf :credit;
    vs:term_status "testing";
    .

:microsite
    a owl:ObjectProperty;
    rdfs:label  "microsite";
    rdfs:comment "Associates a programme to its microsite. For example http://www.bbc.co.uk/
    programmes/b00fm04s and http://www.bbc.co.uk/eastenders/";
    rdfs:subPropertyOf foaf:page;
    rdfs:domain :Programme;
    rdfs:range  foaf:Document;
    vs:term_status "testing";
    .

# Datatype properties

:text
    a owl:DatatypeProperty;
    rdfs:label  "text";
    rdfs:comment "Associates a subtitle event to the corresponding text";
    rdfs:domain :Subtitle;
    rdfs:range xsd:string;
    vs:term_status "testing";
    .

:frequency
    a owl:DatatypeProperty;
    rdfs:label  "frequency";
    rdfs:comment "Associates a channel to its frequency";
    rdfs:domain :Channel;
    rdfs:range xsd:float;
    vs:term_status "testing";
    .
```

```
:position  # should perhaps be changed
  a owl:DatatypeProperty;
  rdfs:label  "position";
  rdfs:comment "The position of a particular series or episode within its containing programme.
    This property can also be used to give the position of an interval within the containing
    timeline .";
  rdfs:domain [
    a owl:Class;
    owl:unionOf (:Episode :Series tl:Interval);
  ];
  rdfs:range xsd:int;
  vs:term_status "testing";
  .

:subtitle_language
  a owl:DatatypeProperty;
  rdfs:label  "subtitle language";
  rdfs:comment "Language of the subtitles emebedded in a particular version";
  rdfs:domain :Version;
  rdfs:range xsd:string;
  vs:term_status "testing";
  .

# Sub-properties of dc:format

:aspect_ratio # Should perhaps be an object property?
  a owl:DatatypeProperty;
    # The following breaks consistency, for some reason
  # rdfs:subPropertyOf dc:format;
  rdfs:label  "aspect ratio";
  rdfs:comment """
    The aspect ration of a particular version.
  """;
  rdfs:domain :Version;
  rdfs:range xsd:string;
  vs:term_status "testing";
  .

:sound_format # Should perhaps be an object property?
  a owl:DatatypeProperty;
  #rdfs:subPropertyOf dc:format;
  rdfs:label  "sound format";
  rdfs:comment """
    The sound format of a particular version.
  """;
  rdfs:domain :Version;
  rdfs:range xsd:string;
  vs:term_status "testing";
  .

# Sub-properties of dc:description

:synopsis
    # The following breaks consistency, for some reason
  #rdfs:subPropertyOf dc:description;
  rdfs:label  "synopsis";
  rdfs:comment """
    The synopsis of a serie, brand or episode.
```

```
    """;
    a owl:DatatypeProperty;
    rdfs:domain :Programme;
    rdfs:range xsd:string;
    vs:term_status "testing";
    .
:short_synopsis
    rdfs:subPropertyOf :synopsis;
    rdfs:label "short synopsis";
    rdfs:comment """
      A short synopsis of a serie, brand or episode.
      Sub-property of po:synopsis.
    """;
    a owl:DatatypeProperty;
    vs:term_status "testing";
    .
:medium_synopsis
    rdfs:subPropertyOf :synopsis;
    rdfs:label "medium synopsis";
    rdfs:comment """
      A medium synopsis of a serie, brand or episode.
      Sub-property of po:synopsis.
    """;
    a owl:DatatypeProperty;
    vs:term_status "testing";
    .
:long_synopsis
    rdfs:subPropertyOf :synopsis;
    rdfs:label "long synopsis";
    rdfs:comment """
      A long synopsis of a serie, brand or episode.
      Sub-property of po:synopsis.
    """;
    a owl:DatatypeProperty;
    vs:term_status "testing";
    .

# Sub-properties of dc:date

:schedule_date
    a owl:DatatypeProperty;
    rdfs:label "schedule date";
    rdfs:comment """
      The schedule date of a broadcast event.
    """;
    rdfs:domain :Broadcast;
      # The following breaks consistency for some reason
    # rdfs:subPropertyOf dc:date;
    rdfs:range xsd:date;
    vs:term_status "testing";
    .

# Sub-properties of mo:duration

:duration
    a owl:DatatypeProperty;
    rdfs:label "duration";
    rdfs:comment "The duration of a version, in seconds.";
```

```
    rdfs:range xsd:int;
    vs:term_status "testing";
    .

# Associating a version to a time object, serving as an anchor for temporal annotations

:time
    a owl:ObjectProperty;
    rdfs:label "time";
    rdfs:comment "Associates an episode's version or a version's segment with a temporal interval.
        This interval can be associated with a timeline, serving as an anchor for further temporal
        annotations, e.g. subtitles or played track";
    rdfs:domain [ a owl:Class; owl:unionOf (:Version :Segment) ];
    rdfs:range tl:Interval;
        vs:term_status "stable";
    .


# External vocabularies (for documentation purposes)

tags:tag
    a owl:ObjectProperty;
    rdfs:label "tag";
    rdfs:comment "Associates an episode to a particular tag";
    rdfs:domain :ProgrammeItem;
    rdfs:range tags:Tagging;
    vs:term_status "stable";
    .

# Giving types for external URIs used
mo:Track a owl:Class .
foaf:Person a owl:Class .
foaf:Agent a owl:Class .
foaf:Organization a owl:Class .
foaf:Document a owl:Class .
tags:Tagging a owl:Class .
geo:SpatialThing a owl:Class .

foaf:based_near a owl:ObjectProperty .
foaf:page a owl:ObjectProperty .
dc:contributor a owl:ObjectProperty .
dc:creator a owl:ObjectProperty .
dcterms:isPartOf a owl:ObjectProperty .
dcterms:hasPart a owl:ObjectProperty .

dc:date a owl:DatatypeProperty .

<http://www.aelius.com/njh#me> a foaf:Person .
<http://metade.org/foaf.rdf#me> a foaf:Person .
<http://moustaki.org/foaf.rdf#moustaki> a foaf:Person .
```

Listing 6: Programmes Ontology specification