Deliverable 5.2 LinkedTV front-end: video player and MediaCanvas API

Jechiam Gural

Pieter van Leeuwen

David Ammeraal

Daniel Ockeloen

11th October, 2012

Work Package 5: LinkedTV Platform

LinkedTV

Television Linked To The Web

Integrated Project (IP)

FP7-ICT-2011-7. Information and Communication Technologies

Grant Agreement Number 287911

Dissemination level[1]	*PU*
Contractual date of delivery	*30th September 2012*
Actual date of delivery	*11th October 2012*
Deliverable number	*D5.2*
Deliverable name	*LinkedTV front-end: video player and MediaCanvas API*
File	*LinkedTV_D5.2.doc*
Nature	*Prototype*
Status & version	*Corrected after QA, Version 1.5*
Number of pages	*32*
WP contributing to the deliverable	*WP 5*
Task responsible	*Noterik*
Other contributors	*Jechiam Gural, Pieter van Leeuwen, David Ammeraal, Daniel Ockeloen* *Noterik*
Author(s)	*Jan Thomsen, Rolf Fricke (Condat)*
Reviewer	*Stéphane Dupont, U Mons*
EC Project Officer	*Thomas Kuepper*
Keywords	*MediaPlayer, Media Fragments, HTML5, SmartTV*
Abstract (for dissemination)	This Deliverables describes the design principles and current status of the LinkedTV Media Player, developed by Noterik B.V., Amsterdam. Technically the Media Player is based on state-of-the-art technology such as HTML5 with Canvas and Video, web-sockets as well as RESTful interfaces. Additionally

[1] • PU = Public
• PP = Restricted to other programme participants (including the Commission Services)
• RE = Restricted to a group specified by the consortium (including the Commission Services)
• CO = Confidential, only for members of the consortium (including the Commission Services))

	it has to implement the W3C MediaFragment 1.0 API in order to be media fragment compliant and it is designed to work on top of the LinkedTV Platform backend. From the functional and user interface side, the Media Player is developed according to the interaction and navigation principles as designed by LinkedTV partner CWI with a specific focus on realizing the functionalities as required by user scenarios or interactive functionality from partners Sound and Vision, rbb and University of Mons.

Table of contents

List of Figures

List of Tables

1 Introduction

The following Deliverable describes the first version of the LinkedTV Media Player as well as of the Media Player Canvas API, which provides the access methods for the display of annotated media content. The LinkedTV Media Player is the central user interface component which realizes and demonstrates all functionalities and user scenarios developed within LinkedTV. Although based on current state-of-the-art and even more advanced web technologies, it is mainly designed with a TV viewing context in mind, rather than a PC based viewing context where keyboard and mouse devices are available. Second, specific to LinkedTV, the LinkedTV Media Player has to be developed as being compliant to the W3C Media Fragment URI 1.0 standard.[2]

The LinkedTV Media Player is developed by LinkedTV Partner Noterik BV. It needs to support a variety of media resources as well as different user interaction models. Its design requires an approach that allows the media player to support different viewing scenarios and user initiated interaction models. The design process of the LinkedTV player is based on the emerging HTML5 standard, which supports playback on a variety of (mobile) devices and provide a reference model for additional HbbTV implementation. The LinkedTV player development strategy incorporates multiple iterations to quickly test prototypes and obtain user feedback to allow additional refinements. In a later stage of the project the LinkedTV HTML5 player is planned to be ported by LinkedTV partner CONDAT to support the HbbTV standard.

The interaction models of the LinkedTV player is based on REST API's, complemented by JSON. There are basically two communication modes between the player and the backend, (1) Player callbacks and (2) LinkedTV scenario instructions. Callbacks provide individual user actions as well as statistical data with respect to the player behaviour, mainly triggered by user behaviour. The LinkedTV scenario instructions send "calls" to the player through the backend of the LinkedTV system to modify the interface and the content on the basis of personalisation and other filtering algorithms.

It is important to note that the playback depends on the algorithms in the LinkedTV platform backend (which are being developed mainly by LinkedTV Work Packages WP2 and WP4). In other words, the player is designed to follow instructions computed in the backend of the LinkedTV system that impose which media or annotations should be shown, filtered out, underlined, restricted within the specific scenario's context.

Media fragments serve as the input for the video player, either in temporal or spatial way. Media fragments are parts of original source videos that are either restricted in time or region, and depending on the instructions will or will not show the original content.

[2] http://www.w3.org/TR/media-frags/

All media needs to be labelled with unique identifiers in the LinkedTV backend. In the display database these unique identifiers are mapped to their corresponding original source videos. These original source videos are stored on the Noterik WebTV platform and are transcoded to a number of streaming formats and support play-out for different resolutions and devices.

The Noterik display database is a caching module that holds both the videos and the metadata in order to manage the presentations and player in efficient ways. Videos can be streamed from the WebTV platform once available in the display database. If an item is initially not in the display database this item is requested along with the metadata and then cached.

One of the challenges in creating a valuable user experience within the LinkedTV context is creating a smart design for the communication interfaces between the LinkedTV player and its corresponding backend system. These should adaptively enrich the end users experience by providing additional media resources to the viewer, but at the same time also preventing the overload of information, which will create stress choice. Displaying or allowing the user to request additional information related to a scene requires new visualisation and interaction strategies. This deliverable describes the first results of the prototype application that was developed.

Contrary to PCs and emerging mobile devices (tablets), most set-top boxes (STBs) still have limited processing power (energy consumption). The current HbbTV 1.1 standard provides very limited options compared to the HTML5 standard. It is difficult to predict how hardware manufacturers will develop new set-top boxes. At this point in time the future HbbTV 2.0 standard is still under discussion,[3] so it is not yet clear which functions developed within the HTML5 environment designed for LinkedTV it will eventually support.

We are currently focussing on a single interface interaction models, while keeping in mind that a second screen application and gestural interaction will be developed in a later phase of the project.

The description of the LinkedTV Media Player is based on the working prototype which was first released in May 2012 and slightly enhanced since.

[3] Cf, e.g. EU-Project HBB-NEXT, http://www.hbb-next.eu

Related LinkedTV deliverables

The design requires the consideration and input of almost all LinkedTV work packages. However, D5.2 strongly relates to the following deliverables in particular:

D3.1 Specification of functionality requirements satisfying user information needs

D3.2 Specification of presentation interfaces for the three scenarios

D3.3 LinkedTV user interfaces sketch

D3.4 LinkedTV interface and presentation engine version 1

D5.1 LinkedTV platform and architecture

History of the document

Table 1: History of the document

Date	Version	Name	Comment
2012/07/5	V0.01	Gural, Noterik	Initial document structure
2012/07/16	V0.02	van Leeuwen, Noterik	
2012/08/5	V0.03	van Leeuwen, Noterik	
2012/08/19	V0.04	Gural, Noterik	
2012/09/20	V1.0	Ammeraal,Noterik	Added 2.1,Added 2.2
2012/09/26	V1.1	Ockeloen	Rewrote 2.1 and 2.2
2012/09/26	V1.2	Thomsen, Condat	Additions throughout the Deliverable added abstract, formatting
2012/10/01	V1.3	Thomsen, Condat	Added Chapter 5
10/10/12	V1.4	Rozendal, Noterik	Changed suggestion from Stephane

1.1 Abbreviations and Acronyms

Table 2: Abbreviations

Abbreviation	Explanation
API	Application Programming Interface
ASR	Automatic Speech Recognition
GIS	Geographic Information System
JARSTOP	Java RDF Stored Procedure

2 Prototype architecture

This chapter describes both the system architecture of the MediaPlayer as the API that can be used to interact with the MediaPlayer.

2.1 MediaPlayer Architecture

The MediaPlayer canvas makes use of a templating system that consists of components which can communicate with each other through means of the so-called "NIC" (short for Network Interface Component[4]). These templates will implement the layout and interaction models of the still evolving user scenario's. The templating system is developed by Noterik and will be published under een open-source licence. Some of the templates are explained in this screencast (scrub to 52 seconds)[5]. By using a template system we can have multiple versions of each prototype in development and incorporate some optional components for the enhanced user control models (like Microsoft Kinect) without too much trouble.

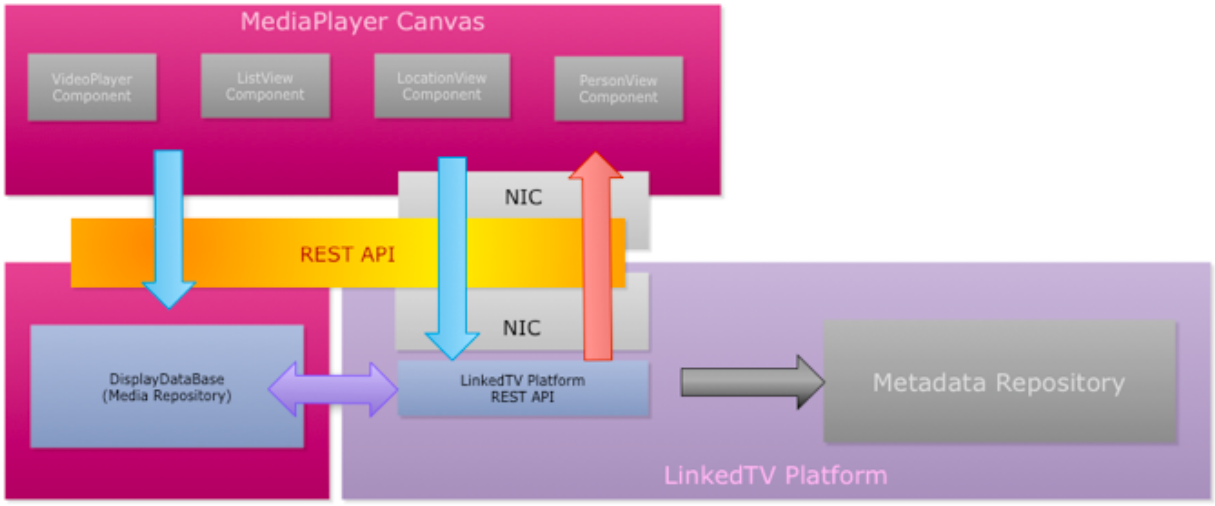


Figure 1: MediaPlayer Architecture

Based on the selected template the player communicates with the display database that stores or gives access to the needed data for each component used. This display database has a RESTful interface and in this way provides the client-side code (browser based) a unified and single point for communication with the backend systems involved in the project. The NIC is responsible for handling the communication between the components

[4] In line with the hardware version NIC is responsible for handeling all communication of the player component with the rest of the system

[5] http://www2.springfieldwebtv.nl/view.html?presentation=/domain/springfieldwebtv/user/admin/collection/8/presentation/2/

and the rest of the platform. During the initialisation phase of a given component, it registers itself with the NIC. The NIC will keep a list of all registered components, and makes them reachable in RESTful way. The NIC is therefore responsible for creating the pathways between components and systems and regulating signals from/to the different parts like user interactions, alternative control methods, video stream events. Most if not all of the outgoing and incoming pathways will be going to the backend maintained by CONDAT who has the master copy of most of the source information generated by the different work-packages and where we need to send our signals from the user actions and statistics to be reused by the system as a whole. We will work with CONDAT in creating these pathways/interfaces using a RESTful interface where needed.

2.2 MediaPlayer API

The video component of the canvas has an extended interface to allow mode direct interaction and feedback systems to attach themselves. It is implemented in HTML5/JavaScript on the client-side of the application. It provides certain basic functions such as play, pause and skip forward, but also allows the user to display active events, select active events, display additional data etc.

The API also provides callbacks to the backend so that user behavior can be analyzed. For example when the user calls the *play()* function, it will also send a callback to the backend describing the interaction, so that these can be analyzed by LinkedTV Work Package 4 Personalisation and Contextualisation. At the moment it is still unclear how the balance will be between this server-side based API's and this client-side API will work out but by providing both we hope to make quick integration and testing possible by not enforcing one model but allowing for both during prototyping and later implementations of the different user scenario's.

3 User Scenario

The existing prototype is being demonstrated based on video material from the TV program "Tussen Kunst en Kitsch"[6] from the Dutch public broadcaster AVRO. This TV format is derived from BBC Antiques Roadshow, a British television show in which antiques appraisers travel to various regions of the United Kingdom (and occasionally abroad) to appraise antiques brought in by local people. It has been running since 1979. In order to better illustrate the functionality to be supported by the LinkedTV Media Player we shortly introduce the scenario description. For a deeper description cf. *D3.1 Specification of Functional Requirements Satisfying User Information Needs*.

3.1 Introducing Rita – the persona

Rita is an administrative assistant at the Art History department of the University of Amsterdam. She didn't study art herself, but spends a lot of her free time on museum visits, creative courses and reading about art. One of her favourite programmes is the *Antiques Roadshow* (Dutch title: *Tussen Kunst & Kitsch*).

Rita likes to watch the Antiques Roadshow because, on the one hand, she learns more about art history, and, on the other hand, because she thinks it's fun to guess how much the objects people bring in are worth. She's also interested in the locations where the programme is recorded, as this usually takes place in a historically interesting location, such as a museum or a cultural institute.

3.2 Rita watches the Antiques Roadshow

Rita is watching the latest episode of the Roadshow. The show's host, Nelleke van der Krogt, gives an introduction to the programme. Rita sees the show has been recorded in the Hermitage Museum in Amsterdam. She always wanted to visit the museum as well as finding out what the link is between the Amsterdam Hermitage and the Hermitage in St. Petersburg. She sees a shot of the outside of the museum and notices that it was originally a home for old women from the 17th century. Intriguing! Rita wants to know more about the Hermitage location's history and see images of how the building used to look. After expressing her need for more information, a bar appears on her screen with additional background material about the museum and the building in which it is located. While Rita is browsing, the programme continues in a smaller part of her screen. After the show introduced the Hermitage, a bit of its history and current and future exhibitions, the objects brought in by the participants are

[6] http://cultuurgids.avro.nl/front/indextkk.html

evaluated by the experts. One person has brought in a golden, filigree box from France in which people stored a sponge with vinegar they could sniff to stay awake during long church sermons. Inside the box, the Chi Ro symbol has been incorporated. Rita has heard of it, but doesn't really know much about its provenance and history. Again, Rita uses the remote to access information about the Chi Ro symbol on Wikipedia and to explore a similar object, a golden box with the same symbol, found on the Europeana portal.[7] Since she doesn't want to miss the expert's opinion, Rita pauses the programme only to resume it after exploring the Europeana content. The final person on the show (a woman in her 70s) has brought in a painting that has the signature 'Jan Sluijters'. This is in fact a famous Dutch painter, so she wants to make sure that it is indeed his. The expert - Willem de Winter - confirms that it is genuine. He states that the painting depicts a street scene in Paris, and that it was made in 1906. Rita thinks the painting is beautiful, and wants to learn more about Sluijters and his work. She learns that he experimented with various styles that were typical for the era: including fauvism, cubism and expressionism. She'd like to see a general overview of the differences of these styles and the leaders of the respective movements.

During the show Rita could mark interesting fragments by pressing the "tag" button on her remote control. While tagging she continued watching the show but afterwards these marked fragments are used to generate a personalized extended information show based on the topics Rita has marked as interesting. She can watch this related / extended content directly after the show on her television or decide to have this playlist saved so she can view it later. This is not only limited to her television but could also be a desktop, second screen or Smartphone, as long as these are linked together. She's able to share this information on social networks, allowing her friends to see highlights related to the episode.

[7] http://www.europeana.eu

Figure 2: Mockup - Standard video playout

Figure 3: Mockup - Menu

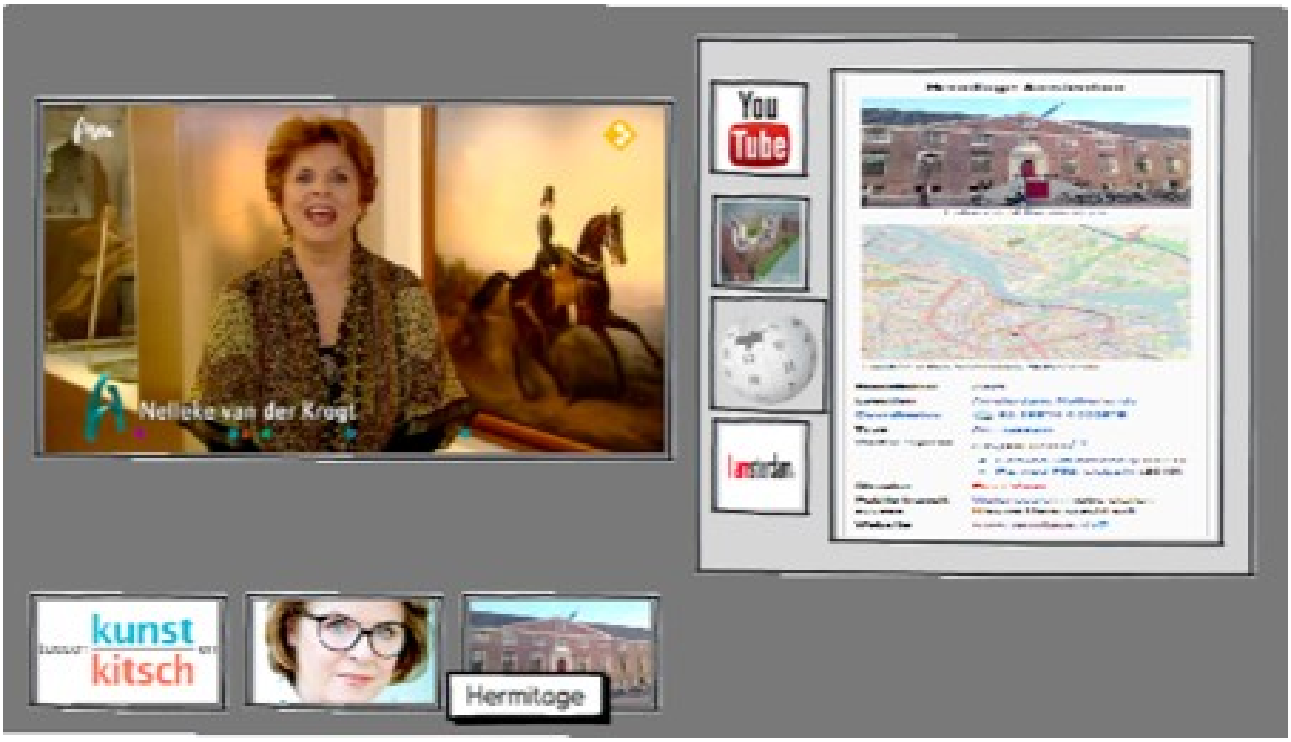


Figure 4: Mockup - Additional information resources

Figure 5: Mockup - Browsing through additional information resources

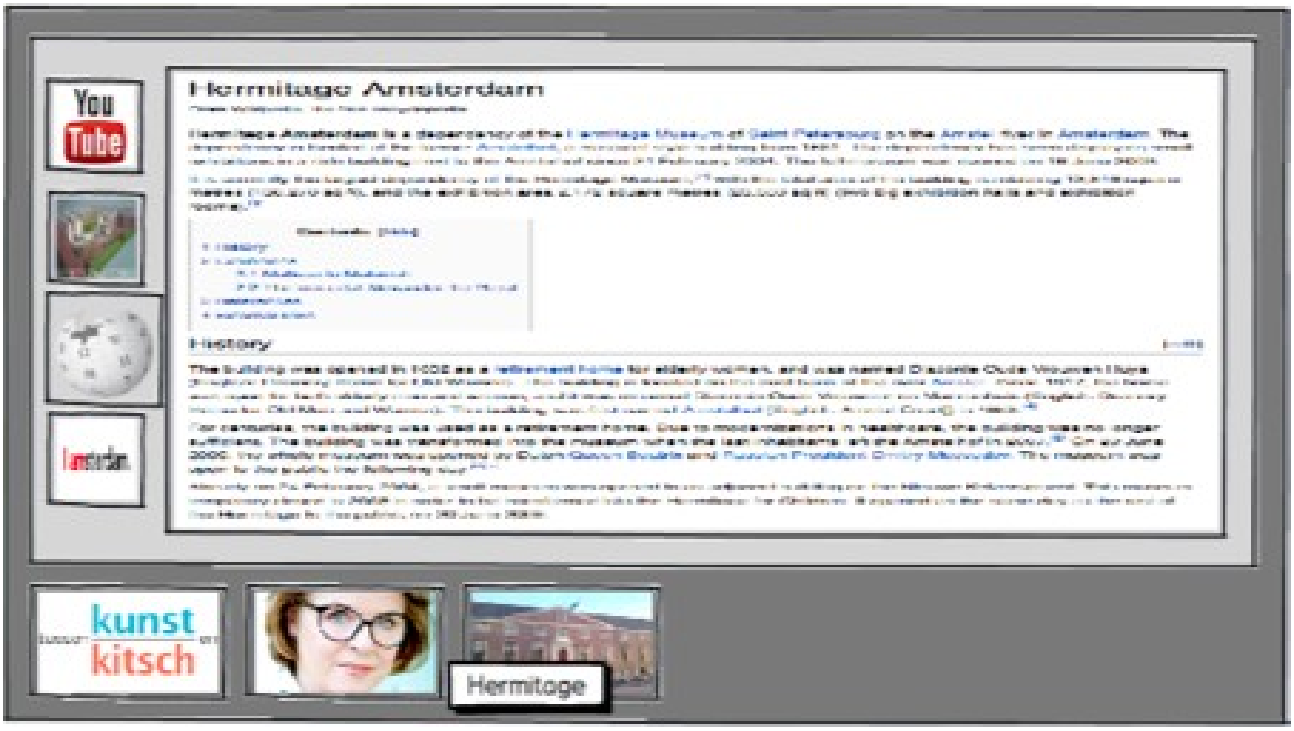


Figure 6: Mockup – Fullscreen information resource

4 Demonstration of the media player prototype

The current status of the LinkedTV Media Player prototype is being demonstrated by means of a video walkthrough. This can be accessed under the following address:[8]

http://www.springfieldwebtv.nl/view.html?presentation=/domain/springfieldwebtv/user/linkedtv/collection/1/presentation/2/

The video walkthrough is not yet made publicly accessible because the digital rights status of the contained material is not yet fully clarified.

In the following the different functions of the LinkedTV Media Player are described as shown in the walkthrough video. The source content is the original TV video material enhanced by additional information which is annotated to different temporal or spatial fragments thereof. Please note that all interaction is done via either remote control or (remote) keyboard.

[8] As long as the digital rights for displaying the material are not finally clarified, access is restricted by password protection. Please use the following access code when prompted after clicking on the link above:

Username: linkedtv

Password: linkedtv12

4.1 Play / Pause

To play or pause the video, press the **P** button on your keyboard or press the **play button** on the remote control.

4.2 Fast forward / backward

To fast forward or backward in video, press the **<** and **>** buttons on your keyboard.

Press the **left** or **right button** for a second on the remote control.

4.3 Show / Hide additional information

To show or hide the additional information in the current scene, press the **S** button on your keyboard.

Press the **menu button** on the remote control.

4.4 Browse additional information items

To browse through the additional information items, press the **left** and **right arrow keys** on your keyboard.

Press the **left** or **right button** on the remote control.

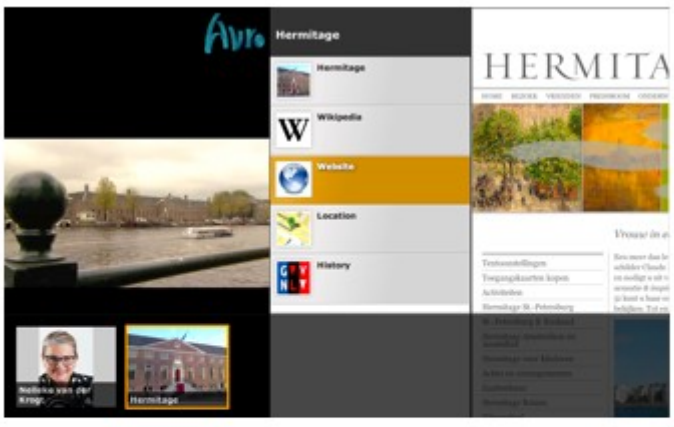
4.5 Select additional information item

To view the additional information about a specific item, press the **enter key** on your keyboard.

Press the **enter button** on the remote control.

4.6 Browse resources

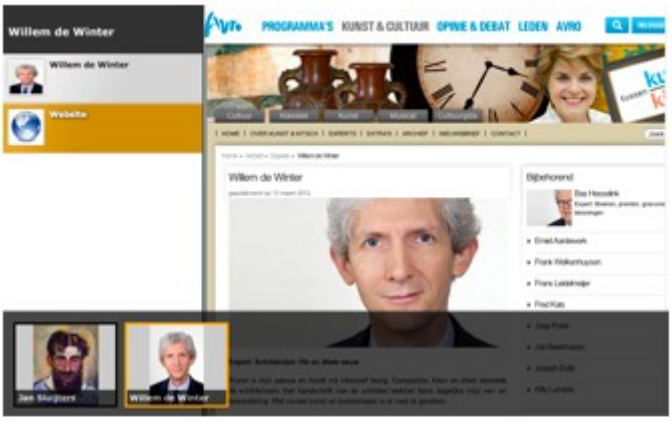
To browse the available resources, press the **up** and **down arrow keys** on your keyboard.



Press the up **or** down button **on the remote control.**

4.7 Full screen resource

To show a resource full screen or return from full screen, press the **F** button on your keyboard.



Press the **menu button** for a second on the remote control.

4.8 Tagging a scene

To tag a scene, press the **T** button on your keyboard.

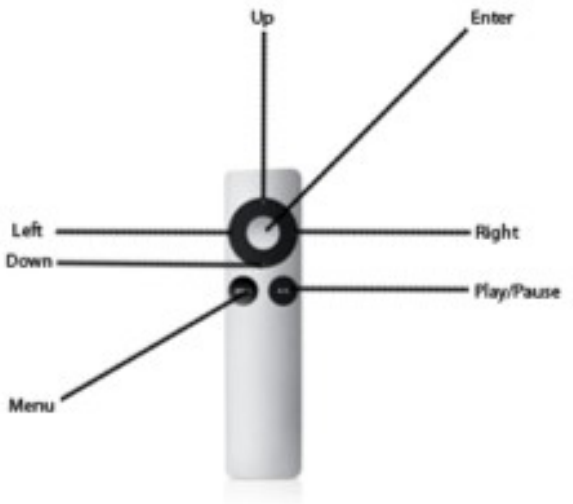
Press the **enter button** for a second on the remote control.

5 Technical Implementation

The prototype application shows the potential of automatically enriching television content for an enriched end-user experience. The chosen scenario allows a wide variety of enrichment techniques to be deployed: techniques to link AV content to Wikipedia articles, named entity recognition and linking of person, location and art style names, feature detection techniques to link close ups of art objects to visually similar objects in the Europeana data set, metadata based linking, etc. For this prototype demonstration the front-end is built for web browsers, using HTML5 and JavaScript for the implementation of the interactive user interface.

This front-end works on top of the existing LinkedTV platform used for the project, an XML based service-oriented platform where audiovisual content is stored, processed into different formats and qualities and made accessible through a RESTful web service. It's capable of storing and manipulating the audiovisual content, metadata and fragment [6] based annotations of the enriched broadcast. In addition, the prototype shows how the linked content can be unobtrusively integrated into a simple but aesthetically attractive TV interface that can be used both during and after the original broadcast, and thus forms a potential to make archived content more attractive.

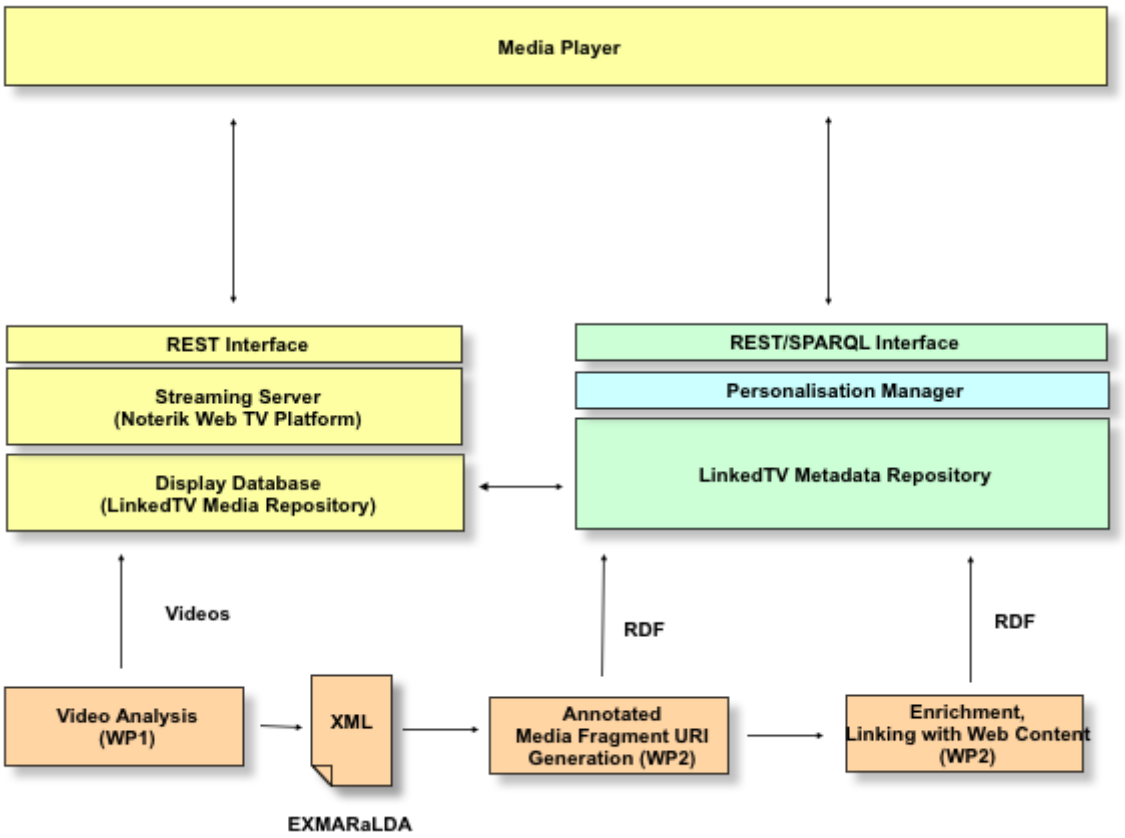
It was possible to control the prototype using a default apple remote control.



5.1 Integration with the LinkedTV Platform

Although it can be used as a stand-alone video player, in order to really generate it's full potential the LinkedTV Media Player frontend, together with the Display Database has to be fully integrated with the LinkedTV platform. There are two main contexts in which this has to be achieved: a) when generating video content for the Display Database and b) when delivering actual annotated media content to the end user.

The following picture shows an overview of the integration within the LinkedTV Platform: [9]



[9] For a description of the whole LinkedTV platform and the overall LinkedTV process cf. *D5.1 LinkedTV Platform*.

5.1.1 Integration of the Display Database

In general, media resources to be analysed, enriched and played may be located anywhere on the Web in all phases of the LinkedTV process. However, in controlled contexts (and the user scenarios defined by LinkedTV include these contexts), the enriched media resources are streamed over the Noterik WebTV platform from the video cache within the Display Database (aka LinkedTV Media Repository). For the Display Database for each original video resource different versions with different encodings and formats for different end user devices are generated.

There are two main interfaces which have to be realized in order to ensure the integration with the LinkedTV platform as a whole:

a) a query and notification interface which tells the Display Database which media resources are actually available for encoding and playout, or in which status they are, respectively. Only those media resources which have finished analysis and annotated media fragment generation process steps are ready for use in a full LinkedTV scenario

b) an update interface which tells the LinkedTV Metadata Repository which new versions of an original media resource have been generated. This is needed because the annotated media fragments generated by Work Package 2 services contain references only to the original media resources, and therefore the platform has to keep track of all different versions.

Both interfaces are realized as part of the LinkedTV Media Layer REST API described in more detail in *D5.1.Linked Media Platform*.

```
GET /resources?status=MF_READY
// get all media resources of which annotated media fragments have been
// generated. Without parameter this service lists all available media
// resources

GET /resource/{id}/status
// get the status of a specific mediaresource

GET /resource/{id}/uris
//get all versions of a specific media resource

PUT /resource/{id}
// set new metadata for a specific mediaresource, in particalur new
versions
// together with information on encoding, format, etc. Values are submitted
// in the body.
```

5.1.2 Integration of the MediaPlayer

When streaming actual content which has been annotated through the LinkedTV process all annotations and media fragment information is provided via its REST API by the Linked Media Layer. In order to support this the REST API provides the following services:

```
GET /resource/{id}/fragments
// get all media fragments of a given mediaresource

GET /resource/{id}/{name}
// get specific metadata of a given mediaresource; e.g. duration, format,
title
// without a {name} all existing metadata are returned

GET /fragment/{id}/{name}
//returns the metadata identified by name associated with a specific
// mediafragment, e.g. hasStart, duration, annotations, region

GET /fragments?{params}
// get all media fragments with a specific property, e.g. annotated with
// certain entities; the conditions are submitted in the body. With params
// the format of the conditation can be specified, e.g. xml, json, sparql
```

6 Current Status and Future Work

6.1 Current status

The prototype was released on May 2012 and will be made available to a selected group of potential users. For a more detailed description of the Media Player Presentation Layer cf. *D3.4 Interface and Presentation Engine*. Subsequent evaluations will be carried out within the context of LinkedTV. The focus will be on the usability and on expanding the application with additional metadata information layers. This will focus on usability of adding additional layers of information of TV such as broadcasts, interaction patterns. Based on the outcomes, we will work in a second version.

The ambition of this second version is deployment on a real life setting. In building the prototype application, we found that automatic linking is not perfect and requires moderation by editors of the programme.

A key step will be the developing of the player user interface according to the design as described in *D3.3 LinkedTV user interfaces sketch*.

To alleviate the amount of editing work we want to investigate the possibility of using social media and crowd sourcing in order to involve users in supplying additional data about specific items in the show. Using the effort of the crowd, we aim to improve and correct the available (context) data and also explore ways to visualize the user's perspective on the material. In this process we aim to maximize the quantity and quality of the content and aim to minimize the amount of moderation that is needed to correct the automated and user generated input.

Lastly, subsequent versions will investigate possible ways of involving and engaging users, for instance by creating games or giving the opportunity to make users experts on certain topics.

6.2 HbbTV Variant

For the first phase of the project the plan foresees to implement a HTML5 variant for the browser, because this is the most appropriate front end for interactive TV with non-linear video and a two-way communication.

For the second phase of the project a restricted HbbTV 1.1 implementation variant for TV Sets is planned. The project intends to profit from an integration of HTML5 elements in upcoming HbbTV releases. But the intended uptake of HTML5 for HbbTV is still under discussion by the standardization organisations, since the combined use of TV sets for broadcast and internet access

1. does not allow a direct uptake of the HTML5 video element, due to the fact that Broadcast video is always running and can not be paused or rewinded; the combination of video and HTML is quite different in HbbTV1.1, where the load of HTML pages is triggered by stream events transmitted with the Broadcast protocol.
2. has to consider the limited CPUs power of TV Sets
3. has to take into account, that TV-Sets usually do not have a pointing device.

For a more detailed discussion of the restrictions of the HbbTV variant and possible solutions see “D5.1, 5.4.4.3 Analysis of options for HbbTV client variant”.

LinkedTV plans to assess the features for the UI designed by WP3 after the first year and decide how they could be provided with HbbTV 1.1 and upcoming HbbTV releases in the next project phases.

7 References

1. Minelli, S. (2007). Gathering requirements for multilingual search of audiovisual material in cultural heritage. Proc. of Workshop on User Centricity – state of the art (16th IST Mobile and Wireless Communications Summit), Budapest, Hungary.
2. "Television and the Future Internet: the NoTube project", Lora Aroyo, Lyndon Nixon, Stefan Dietze and NoTube consortium at the Future Internet Symposium (FIS 2009), Sept 2009
3. Oomen, J. and V. Tzouvaras (2007), Providing Access to European Television Heritage. In Ariadne. Issue 53 http://www.ariadne.ac.uk/issue53/ooman-tzouvaras/
4. K-Space NoE, http://www.k-space.eu/
5. Boemie, http://www.boemie.org/
6. Troncy, R., Mannens, E., Pfeiffer, S., and Deursen, D. van. 2012. Media Fragments URI 1.0 (basic). W3C Proposed Recommendation. Available at: http://www.w3.org/TR/media-frags/