Deliverable 1.5 LinkedTV annotation tool, final release

Names of authors

Jaap Blom

Date of Deliverable

October 29th 2014

Work Package 1: Intelligent hypervideo analysis

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 2014
Actual date of delivery	29th October 2014
Deliverable number	D1.5
Deliverable name	*LinkedTV annotation tool, final release*
File	*LinkedTV_D1.5.docx*
Nature	*Report*
Status & version	Final, v1.0
Number of pages	53
WP contributing to the deliverable	*WP1, WP5, WP2*
Task responsible	*The Netherlands Institute for Sound and Vision*
Other contributors	*RBB*
Author(s)	*Jaap Blom* *The Netherlands Institute for Sound and Vision*
Reviewer	*Nicolas Patz*
EC Project Officer	*Thomas Küpper*
Keywords	*video annotations, broadcast editors, interactive television, data curation, user interface*
Abstract (for dissemination)	*Editor tool v2.0; curation of automatically generated annotations; software.*

[1] • PU = Public
• PP = Restricted to other program 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))

Table of contents

1 Introduction

This document reports on the design and implementation of the final version of the editor tool (ET) v2.0, where its purpose is to serve the program editing teams of broadcasters that have adopted LinkedTV's interactive television solution into their workflow. Two of these teams are currently represented in the LinkedTV project, namely the RBB team and the AVROTROS team (formerly known as AVRO).

The main purpose of the ET is to provide a means to correct and curate automatically generated annotations and hyperlinks created by the audiovisual and textual analysis technologies developed in WP 1 and 2 of the LinkedTV project. Without the intervention of human editors to correct this data, there is a reasonable risk of exposing inappropriate, incorrect or irrelevant information to the viewers of a LinkedTV interactive broadcast.

Compared to the previous version of the ET, the main new feature of v2.0 is the possibility to fetch enrichments, i.e., links to possibly relevant external information sources, *on demand*. This means that an editor now not only has the opportunity to choose from pregenerated enrichments, but also has the possibility to query and retrieve additional enrichments while using the ET.

This new feature emerged from interviews with end-users, and after consultation with the LinkedTV consortium, new services were developed in WP2 to support this new approach.

Moreover, resulting from user tests, conducted by the LinkedTV partner CWI, and several consortium discussions, two new concepts, namely *information cards* and *chapter dimensions*, were introduced into the ET. These concepts increase the reusability of the ET for different use cases and narrow the gap between what editors see in the ET and what the audience sees in the LinkedTV player.

Finally, a modest highlight of v2.0 of the ET is that it also aims to be a development toolkit that enables programmers to configure and extend it with functionalities that are tailored for their player based on e.g., the multiscreen toolkit[2]. Note that this is currently a strategy that is not yet perfectly reflected on a technical level, but is already propagated on the ET's new home page and GitHub[3] page in order to notify interested parties.

This report is a follow up of the deliverable "*D1.3: LinkedTV annotation tool, first release"* and mainly focuses on anything that is news compared to v1.0 of the ET.

[2] Developed by Noterik in WP3 for the purpose of building interactive audiovisual applications

[3] Collaborative working platform for hosting (open source) software: https://github.com/

1.1 Document overview

In order to update the user requirements, the direct end-users RBB and the AVROTROS participated in user trials and interviews in order to obtain feedback on the first version of the ET and new insights into the day-to-day workflow of editing teams. The setup and outcome of these trials as well as the following new user requirements are described in Section 2.The two newly introduced concepts, namely *information cards* and *chapter dimensions* are accounted for in Section 3.

Due to the development of different new web services in WP2 and WP3, the architecture of the ET underwent several changes. Section 4 reports on these new services and the resulting new system architecture.

The current features of the ET as well as updates to the interaction design are described in detail in Section 5. Section 6 describes how users of the open source ET GitHub repository can customize and configure the tool for different (interactive) television programs and their respective editing teams. Section 7 presents a summary of the main points described and gives a lookout on future plans. Section 8 presents the bibliography with related deliverables. Finally, Section 9 presents a glossary to enlighten possibly unfamiliar terminology.

1.2 Exclusions

Unlike what was reported in D1.3, v2.0 of the ET does not include any specific feature with the purpose of influencing the input of WP4’s personalization systems. The main reason for this (see also Section 2) is that current users at RBB and AVROTROS did not express an interest in spending time on specifically curating content for different user groups, however the same users did express an interest in LinkedTV’s capabilities of automatic personalization based on curated data.

1.3 Acknowledgements

CWI greatly contributed in sharing results of relevant user trials, defining methodologies and preparing (and attending) focus group sessions held at NISV and RBB. EURECOM and UEP put serious effort in adapting their enrichments services to support the *on demand* approach. Condat helped out in updating the LinkedTV API to ensure a better connection with the ET. RBB perfectly fulfilled the role of end user and provided NISV with useful comments throughout the development of the ET. The LinkedTV consortium as a whole made sure the ET, touching the work of most work packages, was included in the right discussions at the right time.

1.4 History of the document

Table 1: History of the document

Date	Version	Name	Comment
2014/22/04	V0.1	Jaap Blom, NISV	Initial ToC + updated user requirements
2014/29/04	V0.2	Jaap Blom, NISV; Nicolas Patz, RBB	Updated user requirements
2014/22/09	V0.3	Jaap Blom, NISV	First draft after finishing software
2014/16/10	V0.4	Jaap Blom, NISV	Further updates
2014/17/10	V0.5	Jaap Blom, NISV	Further updates
2014/20/10	V0.6	Jaap Blom, NISV	Further updates
2014/21/10	V0.7	Jaap Blom, NISV	QA ready version
2014/27/10	V0.8	Jaap Blom, NISV; Lampis Apostolidis, CERTH	Integrated feedback from WP1 leader
2014/27/10	V0.9	Jaap Blom, NISV; Nicolas Patz, RBB	Integrated feedback from formal QA
2014/27/10	V0.95	Jaap Blom, NISV	Final version
2014/29/10	V1.0	Jaap Blom, NISV; Lyndon Nixon, MODUL University	Final version with integrated feedback from scientific coordinator

1.5 Purpose of the Document

The purpose of this document is to inform the reader about the progress behind the creation of the final version of the ET.

2 User requirements

2.1 Involving the end-users

For this version of the ET, again[4] end-users at RBB and AVROTROS were involved to provide feedback. This time, the aim was to obtain feedback on the following:

- The day-to-day workflow of program editing teams
- The usability of the ET v1.0.
- The quality of the automatically detected entities and hyperlinks

The first and most important point was to find out more about how program teams work and how feasible it is, or under what circumstances it would be feasible for them to incorporate the ET in their day-to-day workflow. The second point was addressed to obtain direct feedback on the usability of the first version of the ET in order to find out which features should be kept and which features should be dropped or improved. For the third point, CWI executed a test with a user group representing the general public, which provided useful insights[5]. Because of this and the limited availability of both the RBB editing staff and the staff of Tussen Kunst & Kitch (TKK), it was decided, at that time, not to go ahead with a practical test specific for program editors. Regardless of their limited time, both groups showed much interest in learning about LinkedTV and the ET.

For future evaluations of the ET v2.0 (see Section 7.1), the limited availability of staff will be taken into account when creating the tests.

For all the above, CWI greatly contributed in defining the evaluation methodologies and the preparation of materials for the focus group sessions, tests and interviews held. Subsequently RBB and NISV made sure to apply the defined methodologies as much as possible with their user groups. The following sub sections describe the setup and outcomes of the evaluation efforts for TKK cultural heritage show and RBB news show.

2.1.1 Tussen Kunst & Kitsch (AVROTROS)

On the 27th of April, NISV and CWI held a focus group session with the program team of TKK on the premises of NISV. The session agenda was as follows:

[4] See D1.3: *LinkedTV annotation tool, first release*

[5] The results on this will be reported in WP3’s final deliverable D3.8: *Design guideline document for concept-based presentations*

- Short presentation of LinkedTV and the processing workflow (the basic steps from analyzing video content, using the ET, and final interactive broadcast)
- Presentation of the main functionalities of the ET
- Presentation of user tests with the LinkedTV second screen application[6]
- Presentation of future plans of LinkedTV

Throughout the presentations the members of the TKK team could ask questions and provide immediate feedback. During the presentations, the NISV and CWI hosts asked the TKK team questions such as:

- whether the presented aspect/feature would fit their current workflow or could be otherwise adapted to be made to fit their workflow
- what features are especially appreciated and what features are particularly disliked
- whether (if applicable) a feature would target a specific part of their audience

The following table lists the feedback that is applicable to the development of the ET.

Feedback trigger	Feedback	Effect on ET/LinkedTV
As a reaction to showing LinkedTV's personalization approach.	"Personalization should be done automatically, because there are not a lot of editors in the team. TKK only have the general public in mind as their audience, so all curation is done for this type of audience."	• The ET should not incorporate specific features for personalization. • The ET does not need to distinguish between target audiences.
In reaction to the chapter editing functionality of the ET.	"The start and end times of chapters are available during montage, so automatic chapter detection is not necessary if this information could be incorporated into the system." "The short segments of the show are also important as it discusses relevant topics as well."	• For TKK the ET needs to be able to load chapter segmentations provided by the AVROTROS. • In case of automatic chapter detection, for TKK the short chapters are also relevant.
In reaction to showing	"We are wondering how many	• The ET should highlight the

[6] Reported in D6.3: *User trial results*

Feedback trigger	Feedback	Effect on ET/LinkedTV
the way an editor uses the ET to select useful automatically detected entities for creating annotations.	entities are detected on average and should thus be inspected by the editors. We can't spend much time on this." "Seeing the location where the episode was filmed is not very interesting [as an entity], but other locations, such as where the art objects were found are."	most relevant (automatically detected) entities and/or hide irrelevant entities.
In reaction to showing the way an editor uses the ET to select automatically detected enrichments/hyperlinks.	"There are way too many enrichments showing in the ET." "Key frames and a short title or keywords are not enough to judge whether a hyperlink is relevant or not." "A great feature would be that the editor selects a number of entities and then the system would look for relevant hyperlinks. In the current situation there are just too many (irrelevant) links." "Some entities are interesting for inside links, such as e.g. 'gold' [show other episodes with gold objects], but not for outside links [a Wikipedia page on 'gold' is not interesting.]" "It's important to be able to find links that are based on current events." "It's important to be able to find links that are surprising to the editor."	• The ET should be able to reduce the amount of (irrelevant) enrichments, in order to reduce the information overload. • The ET should have a feature that allows a user to fetch enrichments on demand (based on a selection of one or more entities).
In reaction to showing the way an editor adds annotations using the ET.	"For other programs [at AVROTROS] there are boost factors for certain concepts to improve recommendation."	• The ET could be extended with a feature for adding boosting values to annotations in order to aid the WP4 recommendation.
As general feedback on	"Currently the editing team does not	• Ideally the ET could fulfill the

Feedback trigger	Feedback	Effect on ET/LinkedTV
the ET as a whole.	add any contextualization to TKK." "TKK does not have an interactive website because there was never a need for it. But it has a potential. The internet editors publish the press release, but do not add more links. The program and top pieces are described. On social media (Facebook) they give some info to get people's attention." "TKK used to have lists of keywords for each chapter in the episode." "There are high quality production photo's available." "The editor should be responsible for finding links in the ET. He/she should be able to select concepts and the system should look for (surprising) links." "Overall: there is too much 'noise' and too many links."	role of simply being a convenient tool for finding contextual information for television programs. • The LinkedTV platform should be able to deal with different forms of (meta)data coming from the production phase of television programs. • The ET should focus on helping editors quickly find relevant (and surprising) links.

Table 2: *Feedback from the AVROTROS*

2.1.2 RBB

On the 5th of June 2014 two editors worked together with a colleague from the RBB LinkedTV team and discussed the ET in its current version. As a reminder of the motivation for showing this tool, first the end user GUI demo was presented again, then the news show was briefly analyzed in the sense that the attendants agreed on the chapters' headlines and the editors had time to use the ET and make comments and give direct feedback. The ET itself was briefly presented and some strategic tips were issued in order to guide the editors, but the general approach was to let them use the tool and note any issue they may find as any tool that would need hours or days of training and coaching would not have a chance to be accepted into everyday workflows.

The following table lists the feedback that is applicable to the development of the ET.

Feedback trigger	Feedback	Effect on ET/LinkedTV
In reaction to the chapter editing functionality of the ET.	"The prefabricated chapters were not correctly marked. Beginnings and endings did not comply with the beginnings and endings of the reports and anchor presentations. So I had to create new chapters on my own which took about 25 minutes and would not have been affordable in every day's business." "The handling of the editor tool is not intuitive enough especially when it comes to naming the chapters. For example most of the new chapters didn't even have a thumbnail. So I couldn't be quite sure whether changes of naming of chapters were automatically saved or not."	• Automatic chapter detection should be improved to accelerate the editing process. • Chapters must carry a key frame thumbnail.
In reaction to showing the way an editor uses the ET to select useful automatically detected entities for creating annotations	"The prefabricated entities were not as useful as hoped. I suppose only a quarter made sense and was useful. So I had to view every single news report again, note catchwords and time codes to create new additional links. That process again took some time, at least 20 to 30 minutes per report."	• No Editor Tool issue.
In reaction to showing the way an editor uses the ET to select automatically detected enrichments/hyperlinks	"Creating additional links means in my opinion too many working steps which would even not be affordable in everyday business. It would be easier and helpful if mark-in and mark-out could be set right beneath the player while watching every single report to create correct time codes. It would also be timesaving if there was an input screen for naming and additional links right beneath the	• Reduce number of steps for creating new annotations and/or links.

Feedback trigger	Feedback	Effect on ET/LinkedTV
	player.”	
In reaction to showing the way an editor adds annotations using the ET.	“In the beginning I was quite uncertain where to put a URL while creating additional links. Also the usage of the "add“-button seemed not that self-explaining to me. Trial and error was the only way to find out, which took again some time.” “Searching and naming of DBpedia-links was also not self-explaining because the input screen was grayed out and seemed deactivated to me. That’s why I also didn’t get the possibility of creating several additional links in one entity-screen.”	• Clearer guidance and/or a manual.
As general feedback on the ET as a whole.	“As I could see umlauts can not be used but unfortunately there is no warning message if you mistakenly do so. So after having mistakenly used umlauts the complete input screen for additional links disappears after saving. So the editor has to do start from scratch.” “A preview would also be very helpful so that the editor can see the final product while working on it and possible errors.”	• Enable use of Umlauts. • Enable preview templates.

Table 3: *Feedback from the RBB editing staff*

2.2 Prioritized user requirements

Table 4 lists the user requirements for v2.0 of the ET, while Table 5 reports the unfulfilled user requirements taken from D1.3 that haven’t made it in v2.0 either.

ID	Requirement	Description	Priority	In v2.0
AUT1	User login.	Users can log into the system using a user name and password.	Must-have	Yes
AUT2	Authorization by user organization.	Users can only access content of their own organization.	Must-have	Yes
OC1	Content for public users.	For public users there it should be possible to test the tool with a cc-licensed video.	Must-have	Yes
HOM1	There should be an attractive home page with useful context information.	The home page should explain LinkedTV and the purpose of the tool.	Must-have	Yes
CH1[*]	The creation and editing of chapters should be more user friendly.		Nice-to-have	Yes
ENT1*	Cluster entities of the same name, to enable quicker visual inspection of the entities.	Currently, when detected multiple times, entities of the same name are displayed for each occurrence.	Must-have	Yes
ENT2*	Group entities of the same (DBpedia) type, to enable quicker visual inspection of all the entities.	Currently entities of the same DBpedia class/type are not grouped together.	Nice-to-have	No
ENT3*	Remove the key frames shown for each entity.	Currently each entity is displayed with a key frame of the corresponding frame in the video; this fills up the screen unnecessarily.	Must-have	Yes

[*] These requirements all refer to updates of existing functionalities from v1.0. Requirements without an asterisk are completely new requirements.

ID	Requirement	Description	Priority	In v2.0
REL1	The enrichments should include links to related material.	Currently the enrichments only consist of links taken from the white list.	Must-have	Yes
ENR1	External and internal links should be clearly distinguishable.	It should be visually clear which links are internal and which are external.	Must-have	Yes
ENR2*	More recent links should be shown on top.	Currently displayed enrichments are not ordered by date.	Should-have	Yes
ENR3	It should be possible to generate enrichments (on the fly) based on the selection of one or multiple entities.	Currently the system only offers enrichments that are based on each single automatically generated entity. In order to get more meaningful enrichments, an editor should be able to select one or more entities for generating enrichments on the fly.	Should-have	Yes
SAV1*	Publish the editor's choices, tags in the LinkedTV platform.	In v1.0 the save data was stored in a separate RDF graph. In v2.0 this data should be integrated in the overall LinkedTV graph[7].	Must-have	Yes
PRV1	Preview play-out of editor's choices.	A user can view a simplified version of the front-end's play-out, based on the editor's choices, by utilizing a video preview mode	Should-have	No
ST1	Track statistics based on the user actions.	Statistics related to the user's behavior with respect to the acceptance,	Should-have	No

[7] See D2.2 *Specification of lightweight metadata models for multimedia annotation*

ID	Requirement	Description	Priority	In v2.0
		rejection and modification of (automatically generated) annotations & hyperlinks are logged for various evaluation purposes.		

Table 4: *User requirements classified with the MoSCoW method and the availability in v2.0*

ID	Requirement	Description
WL1	Edit items in user organization's white list.	A user can edit (create, update or remove) the (seed) URLs in the user organization's white list.
TPL1	Edit template based on the user's organization.	A user can edit the who-what-where template of the user's organization.
REC1	Add user tags to describe the video as a whole.	A user can select tags from a predefined list to globally describe the main topics of the selected video.
REC2	Add user tags to describe individual chapters.	A user can select tags from a predefined list to globally describe the main topics of the selected chapter.

Table 5: *Unfulfilled user requirements from v1.0 that could not be realized it in v2.0 either*

Table 5 shows requirements that did not make it in v1.0 and won't be implemented in v2.0 of the ET either. The reasons for this vary per requirement:

- **WL1**: There is currently no functionality for this in the LinkedTV API. Implementing this in the ET therefore is not possible.
- **TPL1**: The idea of program templates, e.g., one for RBB News and one for TKK, could not be practically worked out into actual features of the system. Moreover the concept of entity meta types, namely 'who', 'what' or 'where' was dropped in the ET v2.0 as it was no longer considered very relevant in combination with the new concept of chapter dimensions (see Section 3.1).
- **REC1 / REC2**: This was implemented in an intermediate version of V1, but because of other more important issues, no model for storing and utilizing the tags in the platform has been worked out.

Even though not all of the desired features have been implemented, all must-have requirements listed in Table 4 have been met. Moreover, because LinkedTV appreciates the need for specific requirements for different (types of) programs, the ET is available (open source) for any interested party to be further enhanced or customized (see Section 6).

3 Updated data model

Based on new insights, mainly following from the user trials in year 3, two new concepts were introduced to the ET data model, namely (chapter) dimensions and information cards. The following sections describe these concepts, which have considerably influenced the functionalities of the ET v2.0.

3.1 Chapter dimensions

The idea of having chapter dimensions came from the need to semantically organize the annotations added to a video (chapter). This need became apparent after the development and user test with the Linked News demonstrator[8] developed by CWI. In this demonstrator the available context information offered to the user, i.e., the video annotations, were all distributed along different “axes” or as they are now called “dimensions”.

Each dimension in this demonstrator had a different intension of use e.g.

- in depth: show related in depth articles
- opinion: show related opinionated articles
- timeline: show the related information in a timeline

With these examples in mind, the idea came to include the notion of dimension into the LinkedTV model[9]. Using dimensions greatly increased the possibility of expressing use case specific semantics in a generic way:

- each dimension should, by clearly defining its semantics, reflect how information added to it should be used in an application
- there is no limit to the amount of dimensions that can be added, thus for each subtle user feature a new dimension could be added

For the ET having dimensions as configurable entities in the software greatly improves the reusability and adaptability of the tool for different use cases (see Section 6).

In the ET dimensions are currently configured as follows:

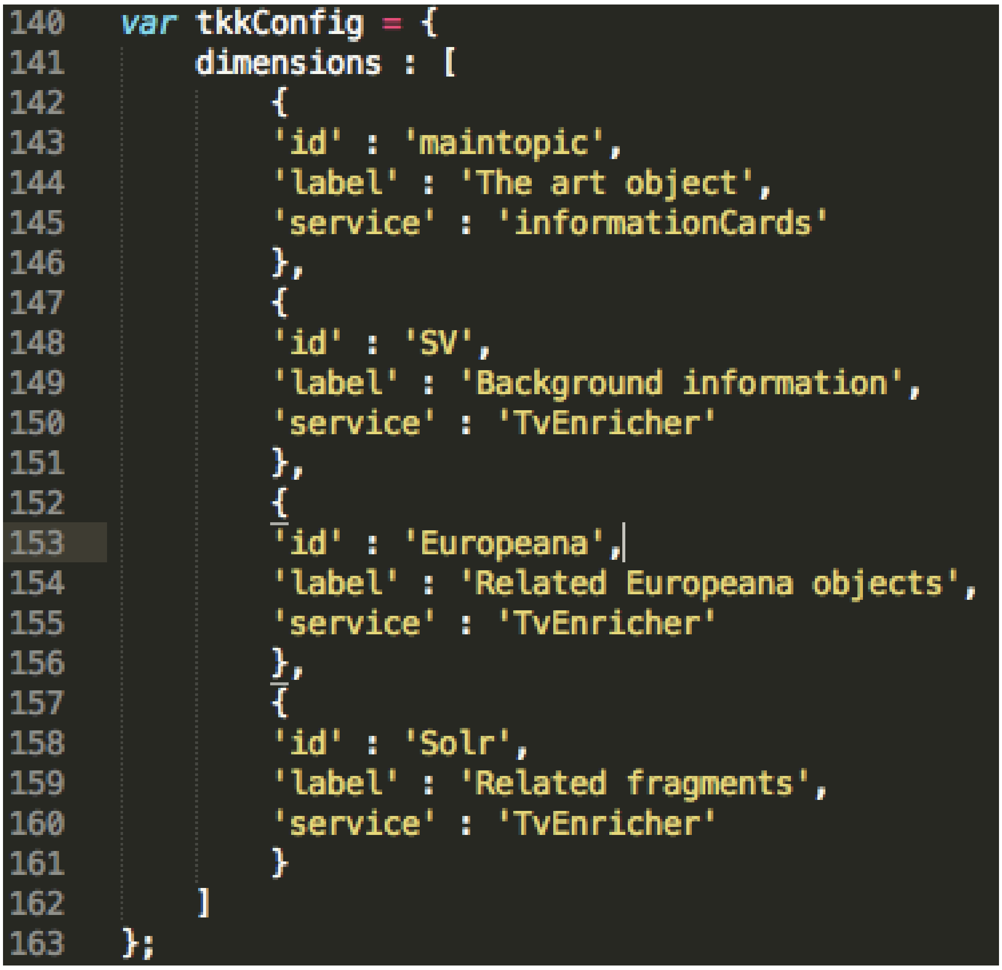
- each program (RBB News, TKK show) needs to define one or more dimensions
    - Each dimension has:
        - an identifier
        - a name (to display in the UI, see Figure 15)

[8] See D3.5: *Requirements Document LinkedTV User Interfaces (Version 2)* for the initial demonstrator and D3.7: *LinkedTV User Interfaces Selected and Refined* for the eventually developed companion application

[9] See D2.2: *Specification of lightweight metadata models for multimedia annotation*

- a web service that handles the fetching of enrichments, currently the following (web) services are supported (see also Section 4):
    - TVEnricher
    - TVNewsEnricher
    - InformationCards[10]

For illustration, the actual configuration looks like this (in JavaScript):



```
var tkkConfig = {
    dimensions : [
        {
        'id' : 'maintopic',
        'label' : 'The art object',
        'service' : 'informationCards'
        },
        {
        'id' : 'SV',
        'label' : 'Background information',
        'service' : 'TvEnricher'
        },
        {
        'id' : 'Europeana',
        'label' : 'Related Europeana objects',
        'service' : 'TvEnricher'
        },
        {
        'id' : 'Solr',
        'label' : 'Related fragments',
        'service' : 'TvEnricher'
        }
    ]
};
```

Figure 1: *JavaScript configuration of the TKK dimensions*

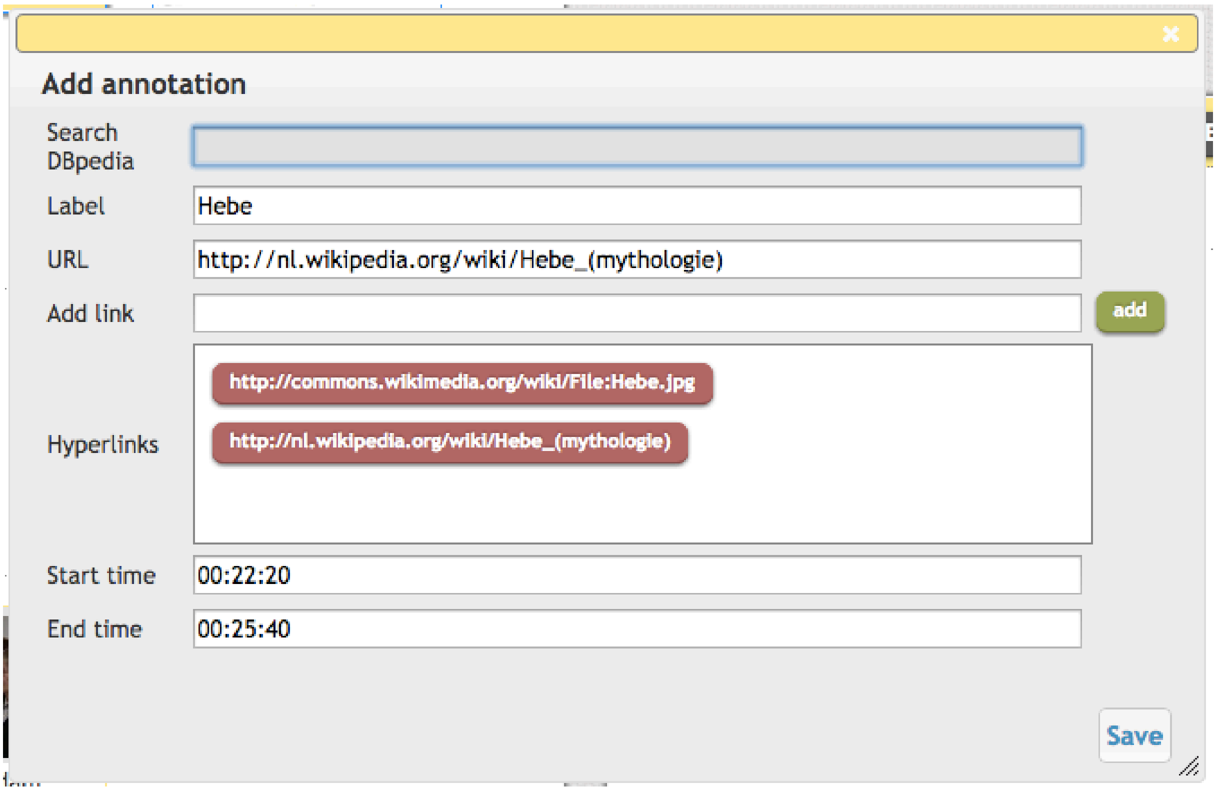
In Section 6.3 the possibility to customize dimensions is addressed in more detail.

3.2 Information cards

In the ET v1.0, the editor was not able to control the information that is shown for entities related to annotations. All an editor could do when creating and annotation was:

[10] The informationCards service is not a separate web service, but rather functionality embedded in the ET (read more in Section 5.5)

- assign a single entity to it (select an auto generated one or add a custom DBpedia URL)
- assign multiple enrichments (select an auto generated one or add a custom URL)



Add annotation

Search DBpedia:

Label: Hebe

URL: http://nl.wikipedia.org/wiki/Hebe_(mythologie)

Add link:

Hyperlinks:
http://commons.wikimedia.org/wiki/File:Hebe.jpg
http://nl.wikipedia.org/wiki/Hebe_(mythologie)

Start time: 00:22:20

End time: 00:25:40

Figure 2: *ET v1.0 showing the annotation edit dialog. The user could only add a single URL representing the (named) entity and add multiple enrichments URLs*

Only able to use the ET this way, the implementation of displaying the actual information related to the entity URL was up to the LinkedTV player[11], which also could only do a dynamic lookup using the entity URL (usually a DBpedia or Wikipedia URL) to fetch the actual information.

Besides needing control over the information shown to viewers, some topics simply cannot be expressed very well using single (named) entities, but actually require the creation of custom entities that could have multiple links to available named entities (expressed as links to e.g. DBpedia).

One such example is the *art object*, which represents the main topic for the TKK videos and fits the aforementioned type of entity:

- most likely there is no online authoritative resource describing the art object shown in the program

[11] Or rather *a* LinkedTV player, using the platform API different player can be implemented

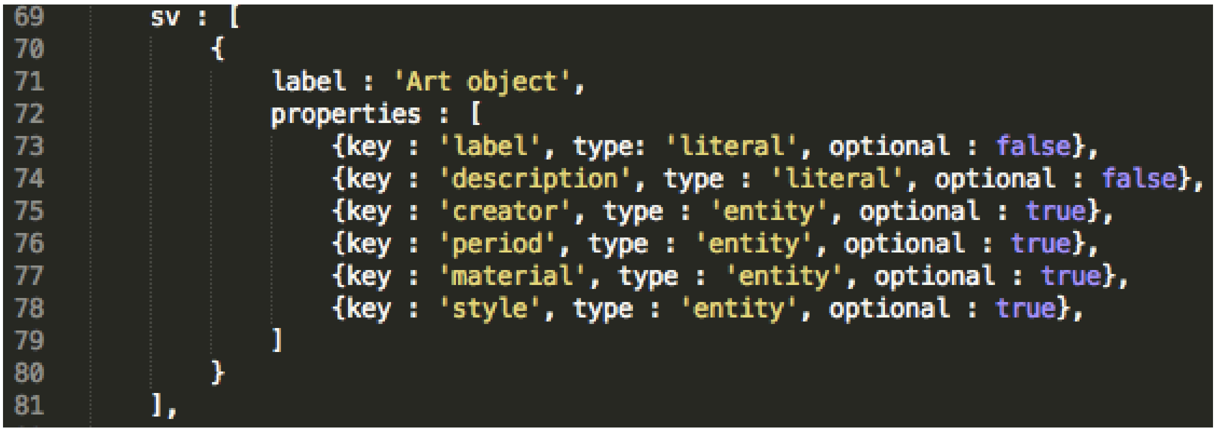
- art objects consist of possibly multiple references to entities that do exist in authoritative resources such as DBpedia or the AAT

So for entities such as art objects, like in TKK shows, or for entities that require a high level of curation, e.g., to ensure high objectivity, which is the case in RBB News, the concept of information cards was conceived.

In the ET these information cards are presented as a set of key/value pairs where the key is the name of the intended property, e.g., “title”, “description”, “creator”, and the value is either a:

- literal (a text string)
- entity, which has:
    - label (a text string)
    - URI (in this version of the ET a URL to DBpedia)

Since information cards could in theory consist of an infinite number of key value pairs, the ET incorporated a templating mechanism to enable configuring templates for editors, aiming to provide a structured way of adding information that can eventually be published to their LinkedTV player. Figure 3 shows the configuration of the art object information card template.



```
sv : [
    {
        label : 'Art object',
        properties : [
            {key : 'label', type: 'literal', optional : false},
            {key : 'description', type : 'literal', optional : false},
            {key : 'creator', type : 'entity', optional : true},
            {key : 'period', type : 'entity', optional : true},
            {key : 'material', type : 'entity', optional : true},
            {key : 'style', type : 'entity', optional : true},
        ]
    }
],
```

Figure 3: *ET configuration – information card template for the TKK art object*

How information cards and their templates are used is described in Section 5.5.

4 System architecture

The architecture described in D1.3 still applies for the most part in v2.0. The biggest changes in the architecture are:

- the incorporation of several new web services developed within WP2 to support enrichments on demand
- the incorporation of the EntityProxy developed in WP3, to support the dynamic inspection of related properties of automatically detected named entities
- the migration of front-end code from Backbone.js[12] to AngularJS[13]

The following sub sections describe the details of these changes.

4.1 Changed dependencies with other WPs

Several new web services have been developed within LinkedTV, which enable the ET to allow users to trigger enrichments based on the selection of a number of entities. This section briefly lists these new services per work package they were developed in. For full details on these services, please consult the specific deliverables they were reported in (which are referenced in the following sub sections).

4.1.1 WP2 new components

TVEnricher

This service is developed by the LinkedTV partner EURECOM and exposes an API that allows the ET, or any other program, to request enrichments on demand, based on a query. In fact, the TVEnricher is a proxy service that passes received queries on to several other services developed in LinkedTV, namely:

- **IRAPI***[14]: a search engine on top of an index created from crawled web documents based on RBB's and TKK's *white list*
- **MediaCollector***: a service that can search through the items published during the past 7 days in 10 different social media platforms
- **LinkedTV SOLR index***: a service that stores information of related video fragments
- **Europeana API**[15]: an API that enables to retrieve items from the Europeana collection

The TVEnricher is configured for both RBB and TKK (see Section 6).

[12] http://backbonejs.org/

[13] https://angularjs.org/

[14] * For more information on these LinkedTV services see D2.6: *LinkedTV Framework for Generating Video Enrichments with Annotations*.

[15] http://pro.europeana.eu/api

TVNewsEnricher

From an ET perspective, the TVNewsEnricher basically fits the same purpose as the TVEnricher; the difference is that it is optimized for the news domain of the RBB use case. Instead of the services listed for the TVEnricher, the Google Custom Search Engine[16] and the Twitter API[17] are used. The result is an API that offers 5 different dimensions to query along:

- **In depth**: news articles that offer in depth information on the specified subject
- **Other media**: news articles from other media sources on the specified subject
- **Opinions**: opinion pieces on the specified subject
- **Timeline**: a timeline comprising of news articles on the specified subject
- **Twitter**: related tweets of the past 7 days on the specified subject

EntityExpansion

The EntityExpansion service (see D2.6), developed by EURECOM, is in the current version not yet fully integrated, but is on the list of useful extensions for the RBB News configuration. Basically this service offers the *on demand* retrieval of additional (named) entities, specifically for the news domain. The idea is that rather than just analyzing the metadata (and subtitles and so on) of the news program itself, it analyses related web documents on the same news topic. This way there is also an opportunity to find entities that come from a broader context, e.g. including yesterday’s news, than just the news item itself. The inclusion of the EntityExpansion service is listed under future work in Section 7.1.

4.1.2 WP3 new component

EntityProxy

The EntityProxy[18], developed by CWI, is a service that takes DBpedia and Wikipedia URLs as input and, based on the supplied DBpedia type, returns a list of related properties from DBpedia.

In the ET this service is used to retrieve information for professional end users when they want to inspect the details of certain automatically detected entities (see Figure 22). Without this functionality the user could only judge the value of an entity by its label. So rather than only reading: “Jan Steen”, by clicking on this label a user can retrieve the additional information, that Jan Steen was a Dutch painter, and read a description of his life and anything else that would be typically displayed in an information box on a related Wikipedia page.

[16] https://www.google.com/cse/

[17] https://dev.twitter.com/rest/public

[18] http://linkedtv.project.cwi.nl/

4.2 Changes to front-end

As mentioned above it was decided to rebuild the front-end for v2.0. The reason for this is that with the addition of dimensions and information cards, the client side data model has changed enough for it to be worthwhile to just rebuild it rather than adapt the old code. Moreover the incorporation of the new *on demand* functionalities required a different approach in (asynchronously) loading data from different services.

Another good reason to rebuild the front-end was to get closer to possible contributors for the ET GitHub, by choosing AngularJS instead of Backbone.js because of its rising popularity and inherent capabilities to create reusable components.

Next to switching to AngularJS, using SASS[19] (in combination with Compass[20]) further improved the possibility to create reusable interface components. SASS is a precompiler for CSS that enables to modularize style sheet code, making it more readable, reusable and extensible for different programmers.

4.3 Changes to the back-end

To support the new *on demand* functionalities, all of the components mentioned in Section 4.1.1 and 4.1.2, namely the TVEnricher, TVNewsEnricher and EntityProxy were integrated into the existing Python Django[21] backend. In this case integration means that the ET back-end handles the HTTP communication between these components ensuring that the user input is translated to valid (component specific) queries and response data is mapped to the ET data model.

Due to ongoing changes and updates of the LinkedTV platform, it was decided to wait with the storage of the new data model to the platform. Instead, all of the editor's efforts are currently stored in the ET itself. For this, a Redis[22] store is used to store all data in JSON format.

Listed under future work (see Section 7.1) is the task for storing all of the curated data in the LinkedTV platform in the form of RDF triples.

[19] http://sass-lang.com/

[20] http://compass-style.org/

[21] https://www.djangoproject.com/

[22] http://redis.io/

5 Functionalities

As a result of the new model (see Section 3) and the new system architecture (see Section 4), and aiming to support the new *on demand* capabilities, the functionalities of v2.0 have changed in several ways.The following sub sections describe these new features in detail and incorporate screenshots for illustration. To increase comprehensibility, the order of the sub sections is based on the recommended workflow for editing a program, which is:

- visit the home page
- login (currently as a RBB or TKK user)[23]
- curate chapters
- per chapter: fetch & curate enrichments along different dimensions

For better understanding, it is recommended to try out the described functionalities using the free trial video in the ET. To do this, open a Chrome[24] browser and navigate to http://editortoolv2.linkedtv.eu/trial. Chrome is needed as it supports the video format currently used by the HTML5 player.

5.1 Home page & logging in

The ET home page can be reached at http://editortoolv2.linkedtv.eu, and informs the visitor about the following:

- What is the ET?
- What is LinkedTV?
- Try out the ET yourselves!
- The ET is open source. Please visit us on GitHub!
- Who are the people behind LinkedTV?
- Special thanks to our external partners

The home page is designed to attract interested parties in finding out more about LinkedTV. Also visitors are given the chance to try out the tool themselves, ultimately hoping that they will be interested contributing to the ET and/or using LinkedTV technology.

The video used for the trial is creative commons[25] licensed and comes from Open Beelden[26], which is a site hosted by NISV.

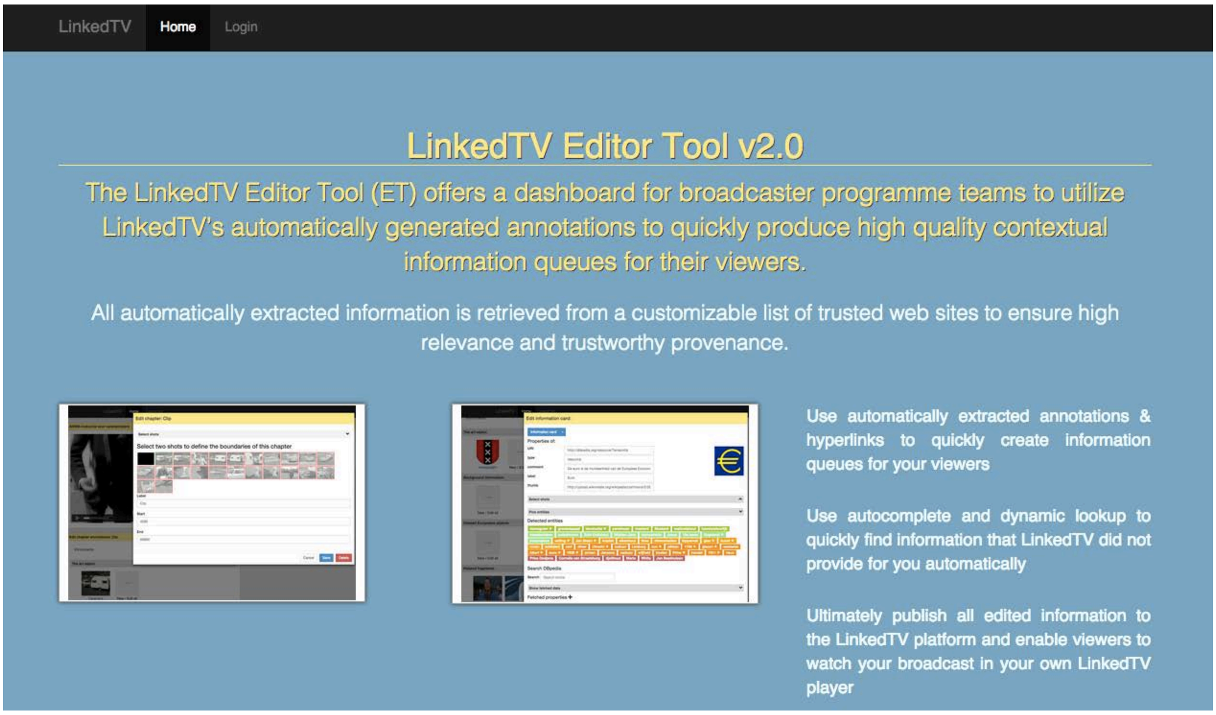
Figures 4-7 show the different sections of the home page.

[23] It is also possible to skip login and go to a free trial page (see Figure 5)

[24] http://www.google.nl/intl/nl/chrome/browser/

[25] http://en.wikipedia.org/wiki/Creative_Commons

[26] http://openbeelden.nl/



LinkedTV Editor Tool v2.0

The LinkedTV Editor Tool (ET) offers a dashboard for broadcaster programme teams to utilize LinkedTV's automatically generated annotations to quickly produce high quality contextual information queues for their viewers.

All automatically extracted information is retrieved from a customizable list of trusted web sites to ensure high relevance and trustworthy provenance.

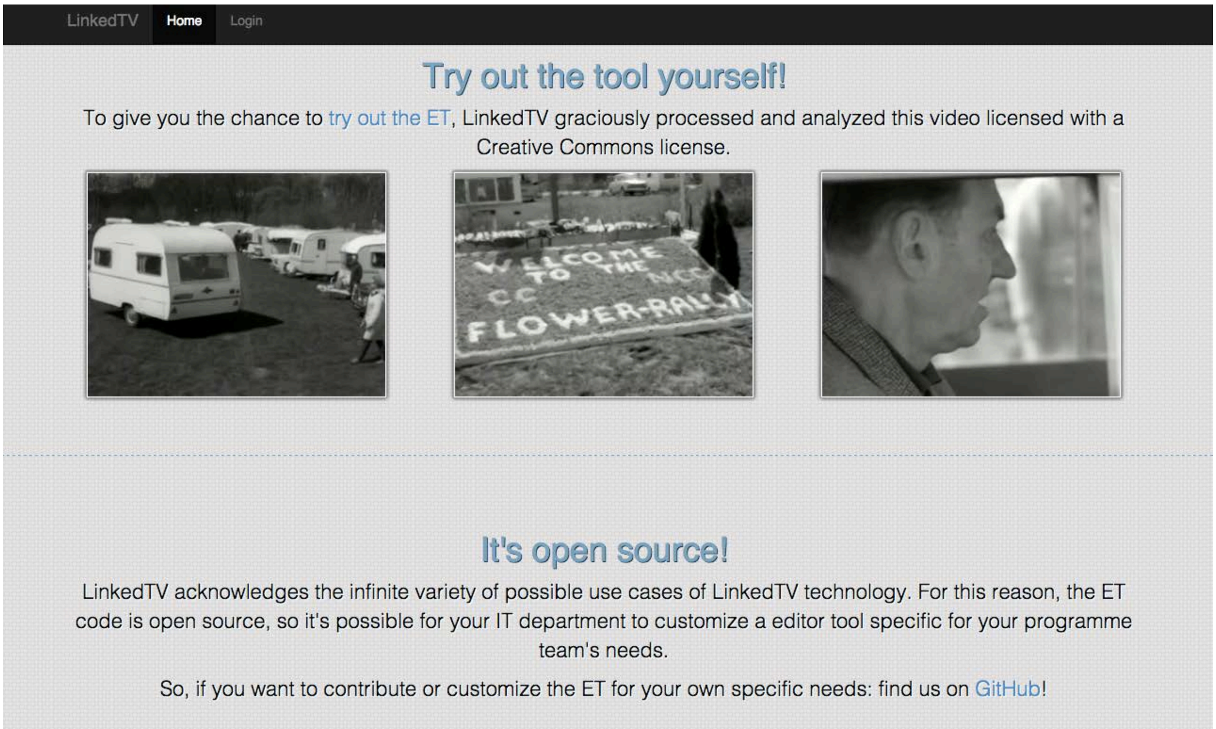
Use automatically extracted annotations & hyperlinks to quickly create information queues for your viewers

Use autocomplete and dynamic lookup to quickly find information that LinkedTV did not provide for you automatically

Ultimately publish all edited information to the LinkedTV platform and enable viewers to watch your broadcast in your own LinkedTV player

Figure 4: *Home page – top section informing about the ET*

The first section informs the visitor about the overall capabilities and purpose of the ET.



Try out the tool yourself!

To give you the chance to try out the ET, LinkedTV graciously processed and analyzed this video licensed with a Creative Commons license.

It's open source!

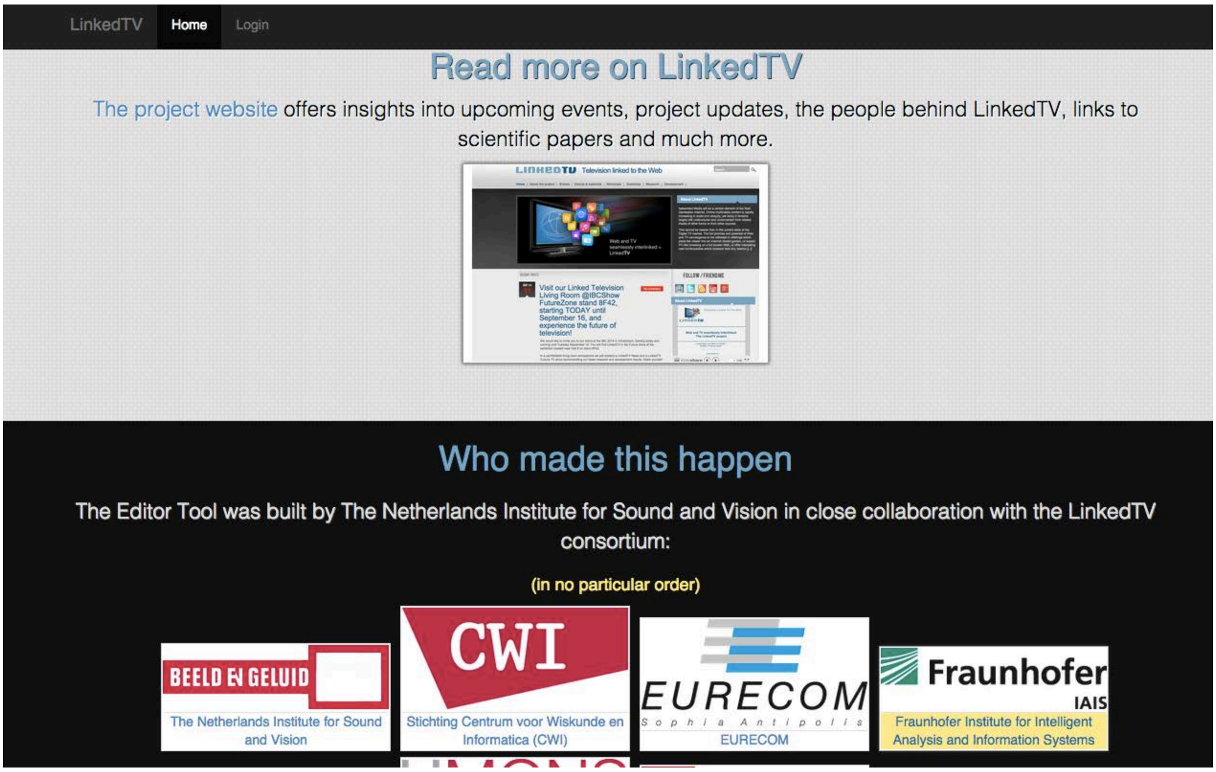
LinkedTV acknowledges the infinite variety of possible use cases of LinkedTV technology. For this reason, the ET code is open source, so it's possible for your IT department to customize a editor tool specific for your programme team's needs.

So, if you want to contribute or customize the ET for your own specific needs: find us on GitHub!

Figure 5: *Home page – link to the trial page and link to the ET on GitHub*

The second section informs the reader about the trial version of the ET.

The third section directs the visitor to the ET GitHub page and informs the reader about the possibilities of customizing the ET for specific use cases.



Read more on LinkedTV

The project website offers insights into upcoming events, project updates, the people behind LinkedTV, links to scientific papers and much more.

Who made this happen

The Editor Tool was built by The Netherlands Institute for Sound and Vision in close collaboration with the LinkedTV consortium:

(in no particular order)

The Netherlands Institute for Sound and Vision

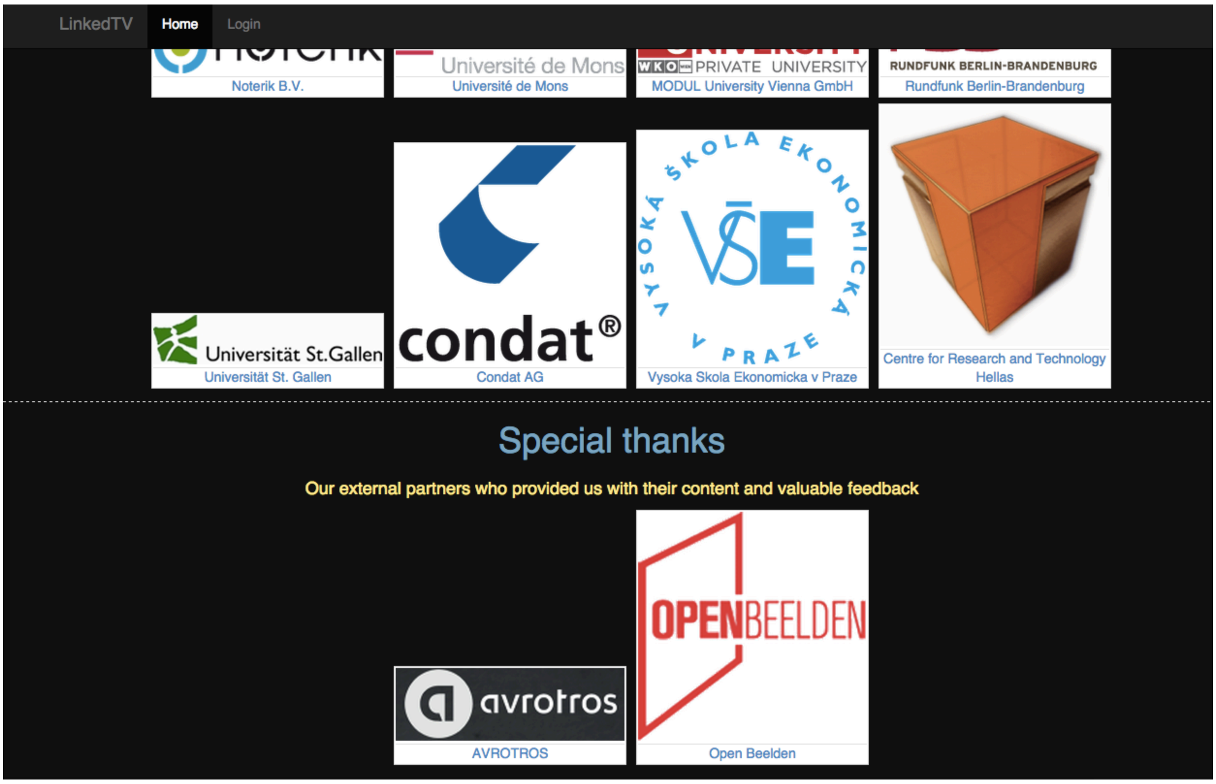
Stichting Centrum voor Wiskunde en Informatica (CWI)

EURECOM

Fraunhofer Institute for Intelligent Analysis and Information Systems

Figure 6: *Home page – link to project website and partner credits*

The fourth section presents a link to the LinkedTV project website.



Noterik B.V.

Université de Mons

MODUL University Vienna GmbH

Rundfunk Berlin-Brandenburg

Universität St. Gallen

Condat AG

Vysoka Skola Ekonomicka v Praze

Centre for Research and Technology Hellas

Special thanks

Our external partners who provided us with their content and valuable feedback

AVROTROS

Open Beelden

Figure 7: *Home page – special thanks section*

The final two sections show the logos of consortium partners and affiliated partners.

For accessing the ET, known users, currently being RBB and TKK, have been authorized to login to the system so they can curate their videos. By clicking “login” in the navigation bar at the top of the page the user is directed to the login page:

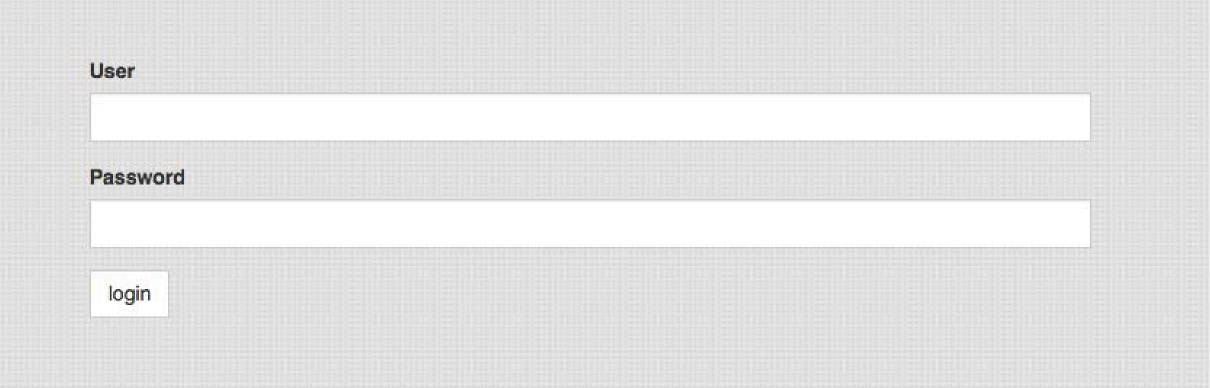


Figure 8: *Login screen*

After successfully logging in, the user is directed to the video selection page, which is described in the next section.

5.2 Choosing a video

The following image illustrates the video selection screen, which shows all of the videos that were successfully processed by the LinkedTV platform for the organization the user is a member of.

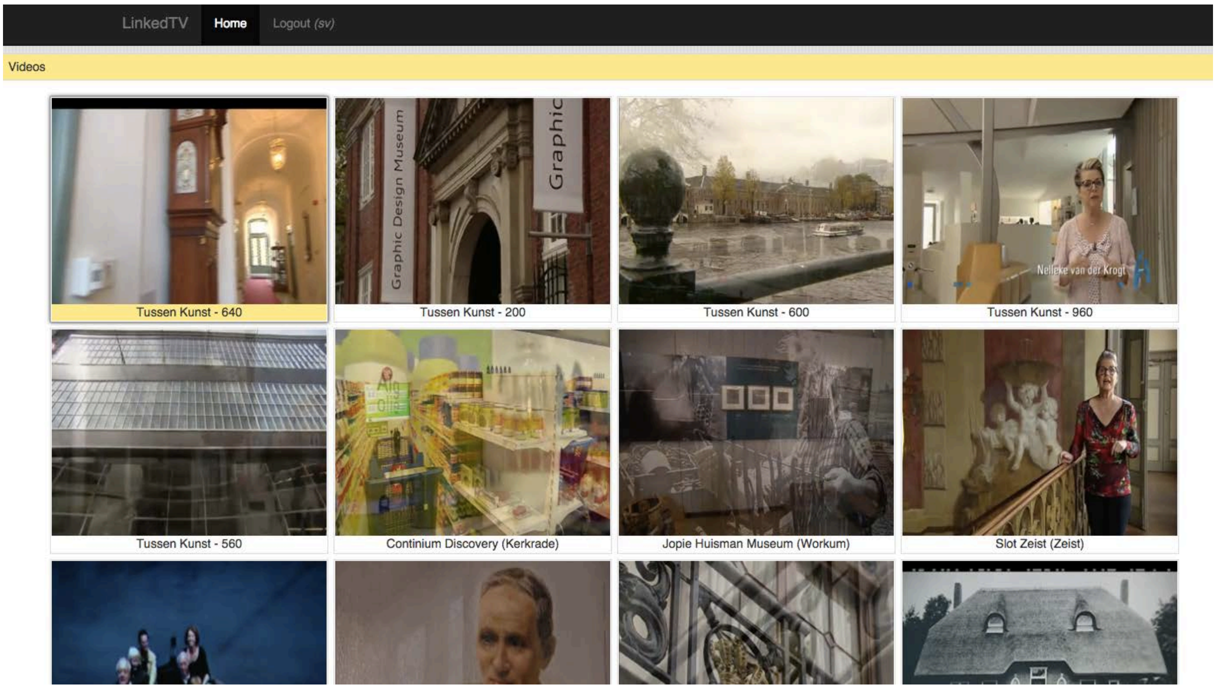


Figure 9: *Video selection page showing a thumbnail and title of each video*

By clicking on one of the video thumbnails, the user is directed to the editing page of the selected video. The following sections describe the different functionalities of this page.

5.3 Curating chapters

The LinkedTV platform offers several means of automatically identifying the different chapters of a video (see Section 3 of D1.4: *Visual, text and audio information analysis for hypervideo, final release*). In most conceivable cases, depending on each television program, a chapter is a part of the program that is about a single topic. For RBB for instance each news item is a separate chapter; for TKK the main chapters are about a single art object.

Before a user starts adding annotations, linking different kinds of enrichments (see Sections 5.5 and 5.6), the possibly available automatically generated chapters, need to be edited in order to make sure they are correct.

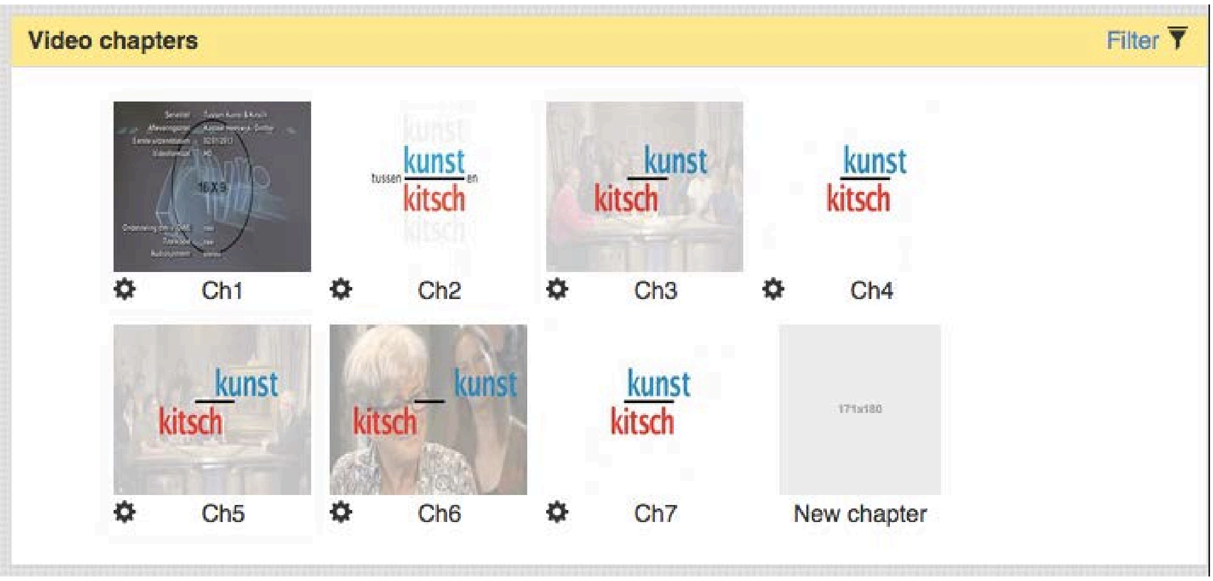


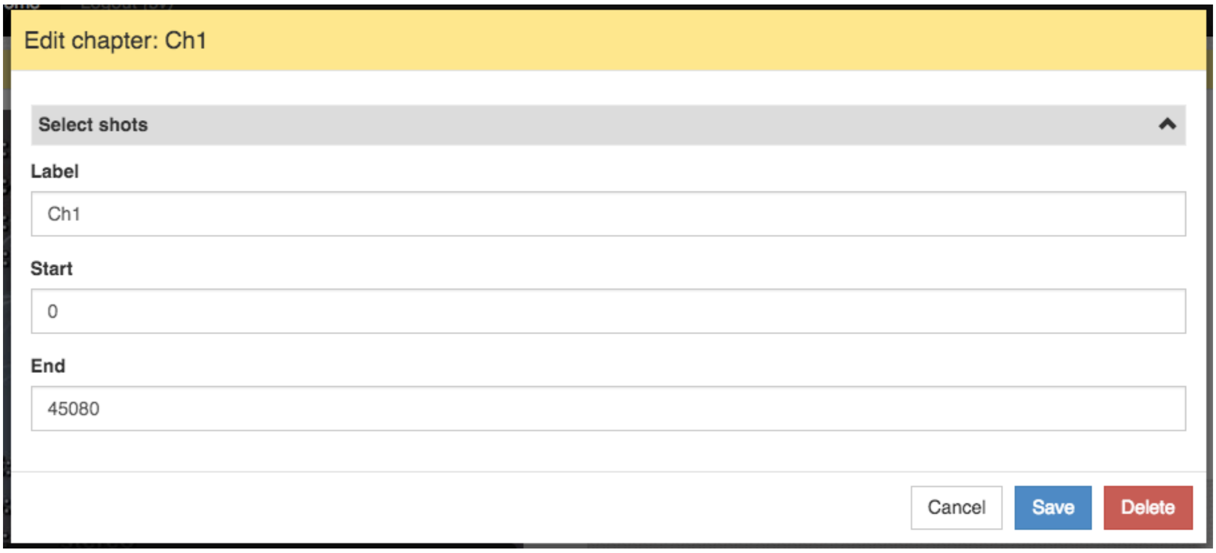
Figure 10: *Chapter panel – showing autogenerated chapters only*

Figure 10 shows the video chapters of a TKK program that has not yet been edited; the cog icon below each chapter thumbnail indicates that a chapter was automatically generated and not edited by a human (curated). In this example it is clear that the chapters were created by the chapter segmentation algorithm, which relies on the TKK bumper re-detection[27], as the TKK logo can be seen in each (except the first) thumbnail, which always reflects the starting frame of the chapter. For correctly editing chapters, the user is recommended to:

[27] CERTH/ITI implemented this algorithm for the TKK chapter segmentation as each main chapter in TKK starts with showing the TKK logo (see also D1.4 in the Bibliography).

- create proper titles for each chapter
- make sure to check the boundaries of each chapter are correct
- add possibly missing chapters if necessary
- remove possibly unnecessary chapters

In order to edit the predefined chapters a user can double click on one of the thumbnails to open the following dialog:



Edit chapter: Ch1

Select shots

Label

Ch1

Start

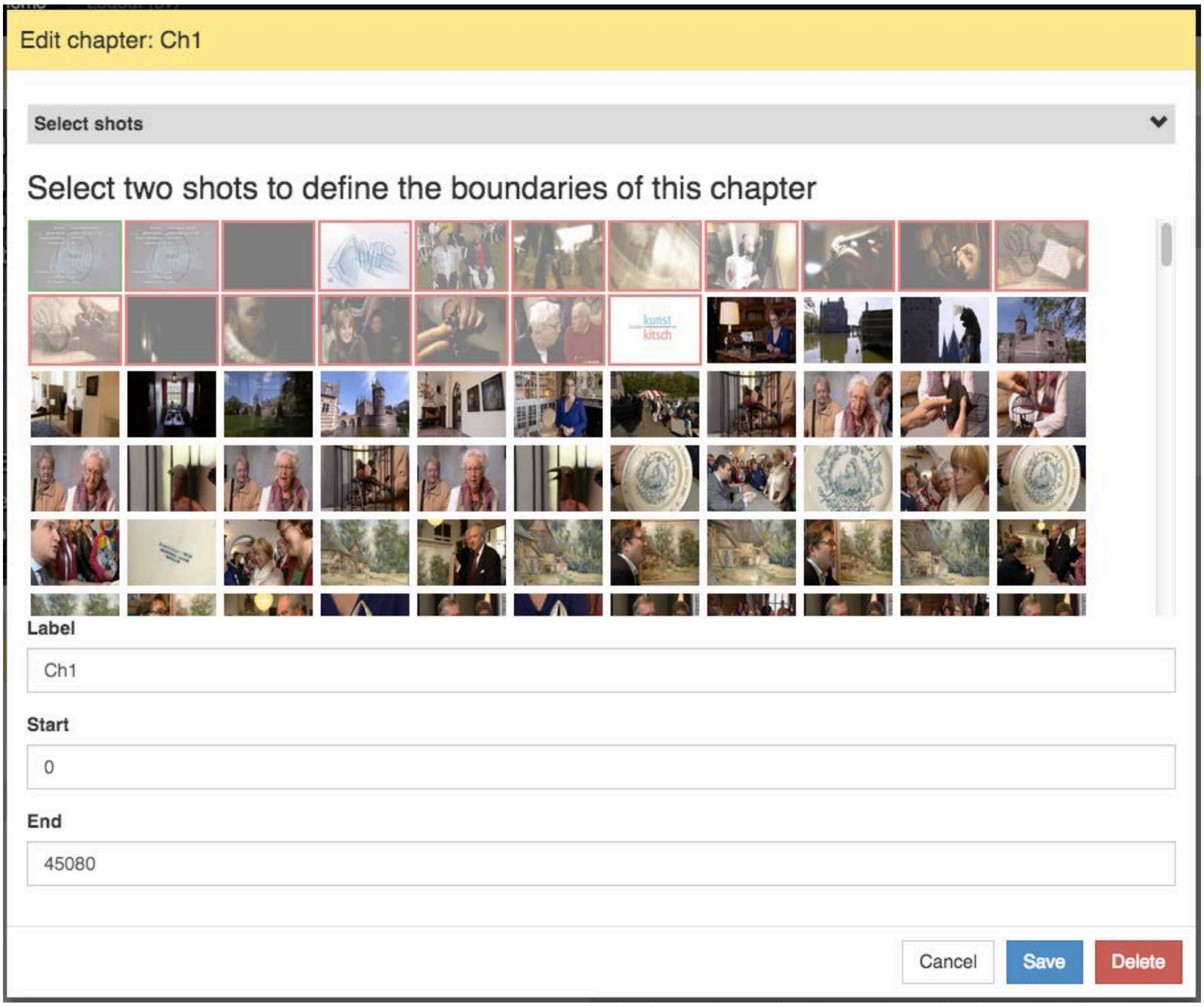
0

End

45080

Figure 11: *Chapter edit dialog – the shot selection is hidden*

When clicking the “Select shots” bar the shot selection is opened:



Edit chapter: Ch1

Select two shots to define the boundaries of this chapter

Label: Ch1

Start: 0

End: 45080

Figure 12: *Chapter edit dialog – showing shot selection; shots included in the chapter are highlighted*

The edit dialog can be used to change the title as well as the start and end times (in milliseconds) of the chapter. As directly entering the amount of milliseconds is not very convenient, the user can select a starting and ending shot by using the shot selection panel. When doing so, the amount of milliseconds is automatically set in the start and end fields. If after saving the start and end is not perfect, the user can fine-tune them by adjusting the values in the corresponding fields.

When a chapter has been edited and saved, the cog icon in the chapter panel is replaced by a ‘human’ icon, indicating that the chapter has been curated (see Figure 13 below).

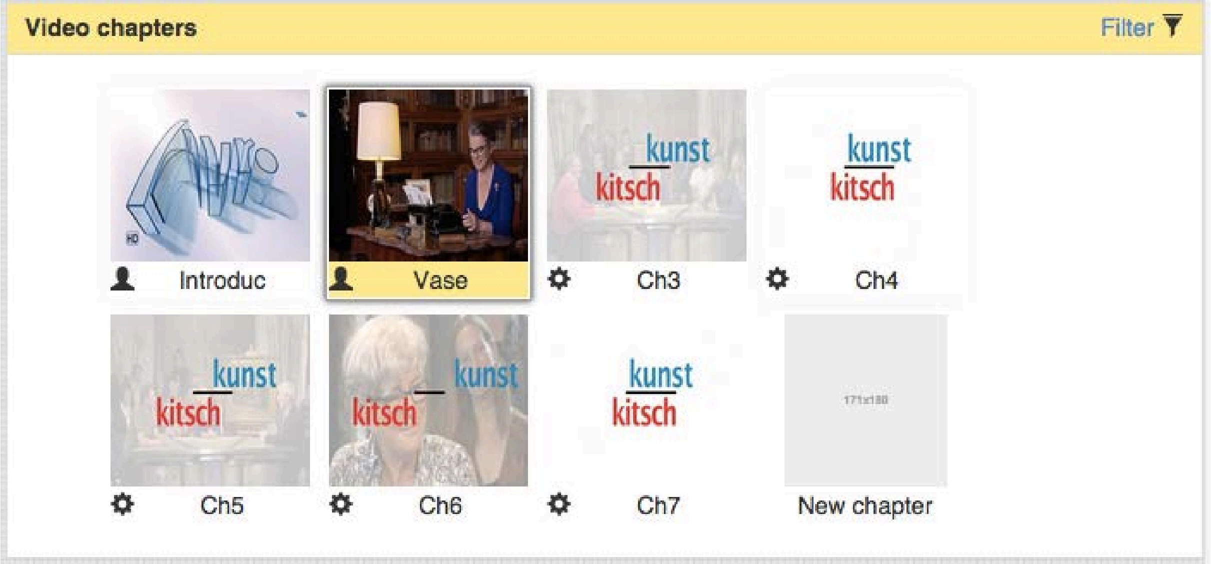


Figure 13: *Chapter panel – showing two curated chapters*

When pressing the delete button (at the bottom, see Figure 12) a chapter can be deleted. Although removed from the view, this will not affect automatically generated chapters to be removed from the LinkedTV platform. When reloading the page, all automatically generated chapters are shown on the page again. To avoid confusing the user, the filter button in the top right of the chapter panel can be used to only show the chapters that were curated (see Figure 14)

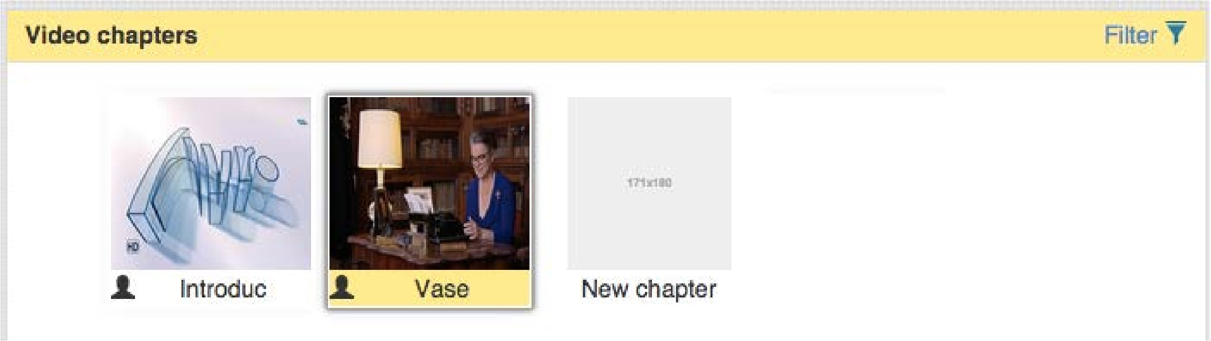


Figure 14: *Chapter panel – showing curated chapters only (filter activated)*

Finally by pressing the “New chapter” icon a new chapter can be added. The same screen as Figure 11 is shown and after saving, the chapter is simply added to the list of chapters.

After a user finishes the editing of the automatically generated chapters, it is time to start adding annotations to each one of them. In this version of the ET there are currently two kinds of information that can be attached to annotations

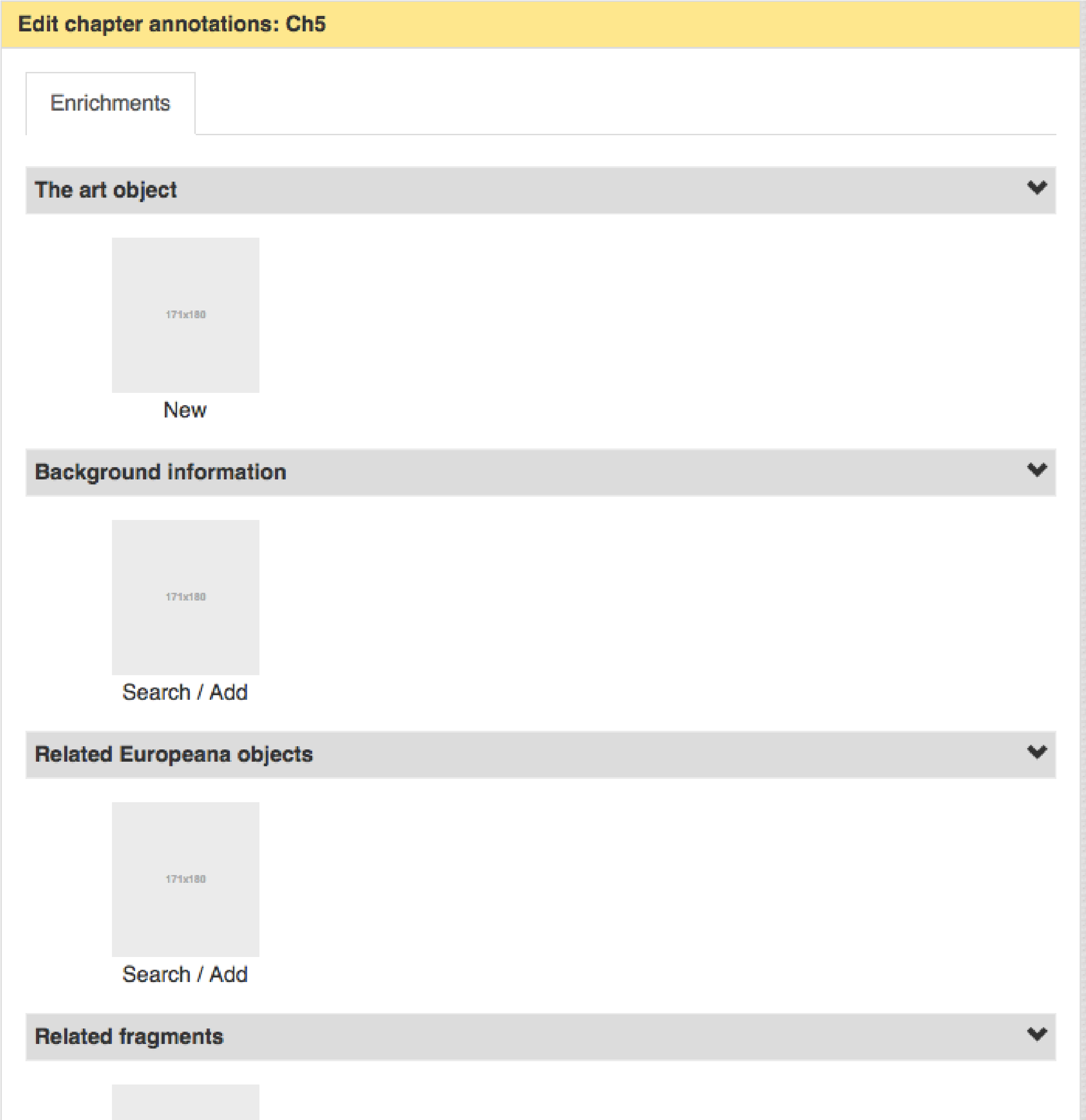
- information cards
- enrichments

Section 5.5 describes everything there is to know about working with information cards and Section 5.6 delves into how to work with all the other types of enrichments. However before going into detail about the actual addition of annotations it is necessary to explain about how these annotations are all going to be related to certain dimensions per chapter.

The following section describes how the user can interact with configured chapter dimensions.

5.4 Chapter dimensions

As described in Section 3.1, each chapter has one or more dimensions configured for adding enrichments to. The result of such a configuration, in this case for TKK, is reflected in the annotation panel when navigating to a chapter (by selecting one in the chapter panel as shown in Figure 10):



Edit chapter annotations: Ch5

Enrichments

The art object

Background information

Related Europeana objects

Related fragments

Figure 15: *Annotation panel – showing the dimensions configured for TKK*

If enrichments already have been added this panel looks like the following:

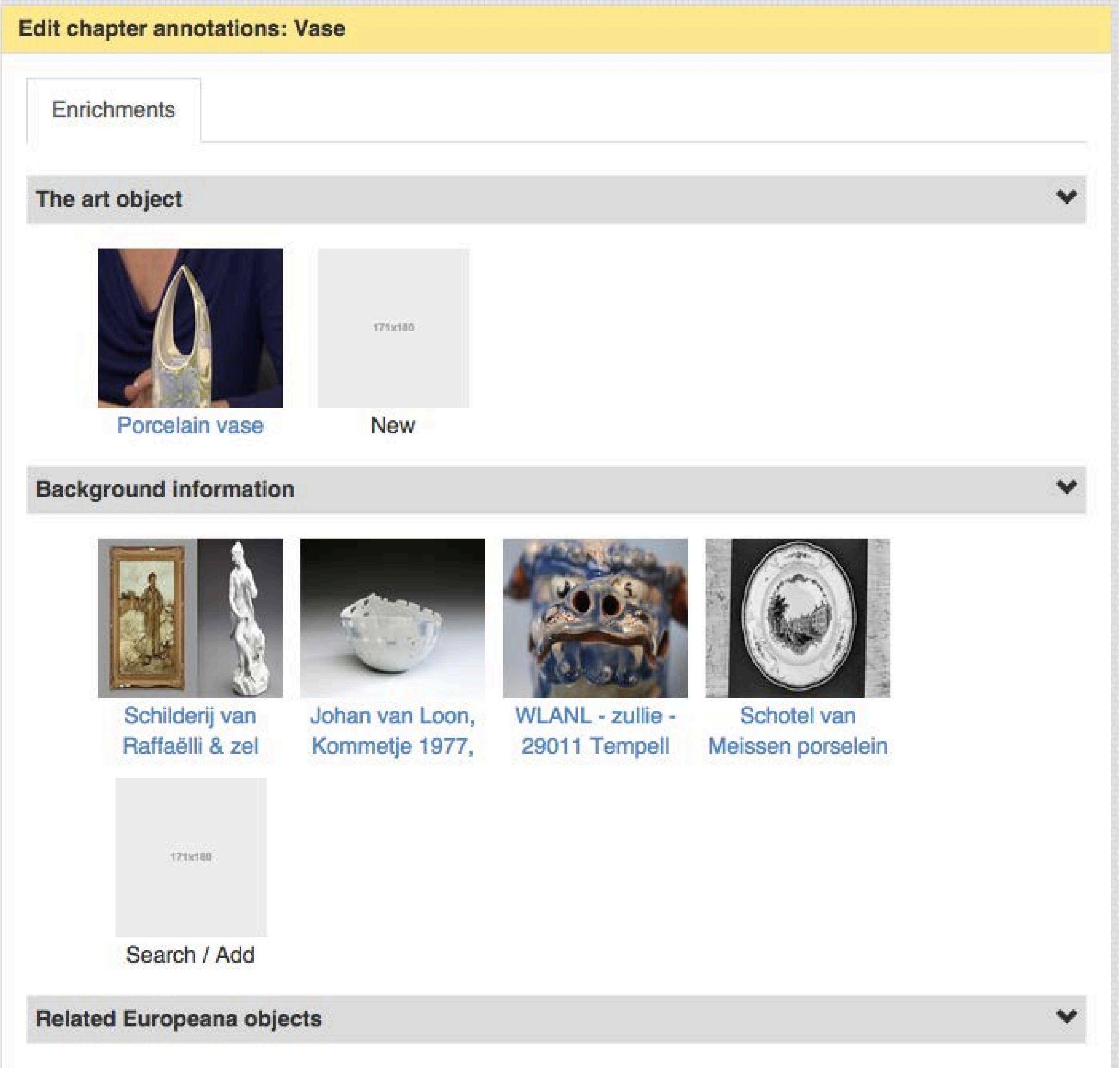
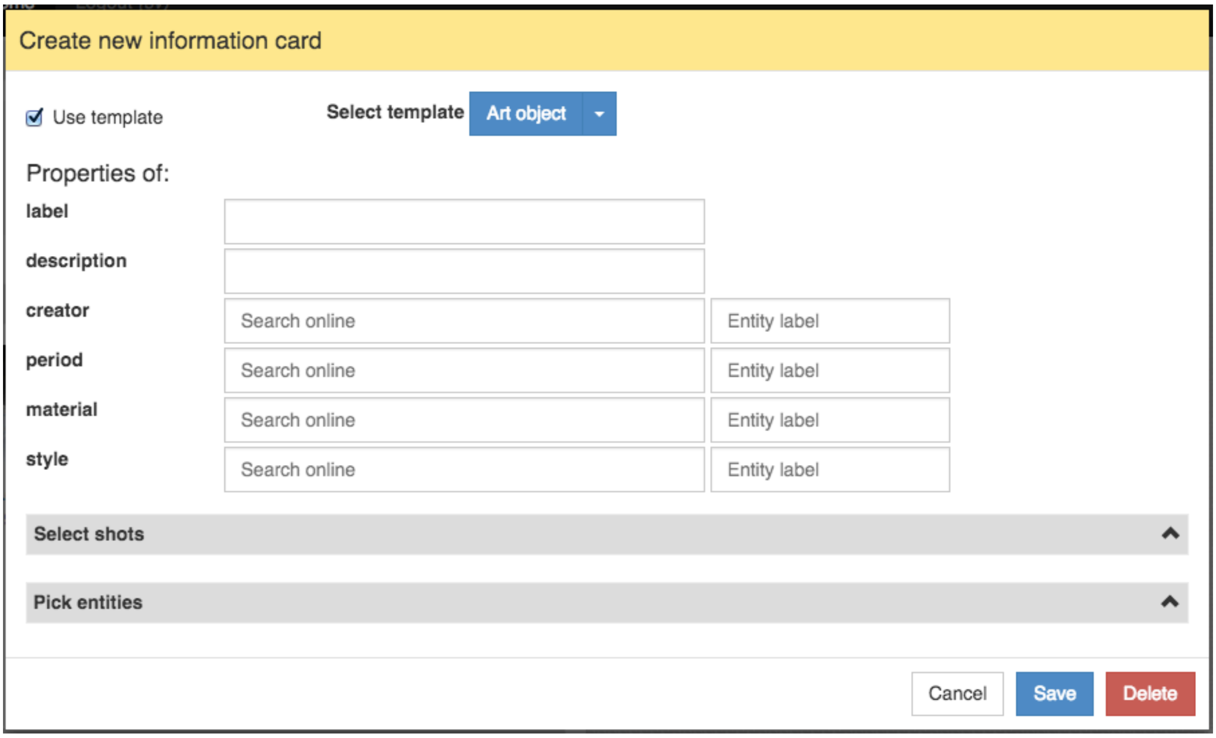

Edit chapter annotations: Vase
Enrichments
The art object
Porcelain vase
171x180
New
Background information
Schilderij van Raffaëlli & zel
Johan van Loon, Kommetje 1977,
WLANL - zullie - 29011 Tempell
Schotel van Meissen porselein
171x180
Search / Add
Related Europeana objects

Figure 16: *Annotation panel – showing two dimensions with saved annotations*

After curating chapters, adding enrichments, in either the form of *information cards* or 'regular' enrichments, is one of the main tasks of the editor. The following two sections describe how these different kinds of enrichments can be added to each dimension.

5.5 Adding information cards

Following Section 3.2, Figure 1 shows that the dimension named 'Art object' is configured to be an information card dimension. The effect of this is that when double clicking the 'New' button in the 'Art object' dimension (see Figure 16) opens an empty information card edit dialog (see Figure 17). When double clicking an existing information card the same screen is opened, but with all of the saved properties filled out in the form at the top (see Figure 18).



Create new information card

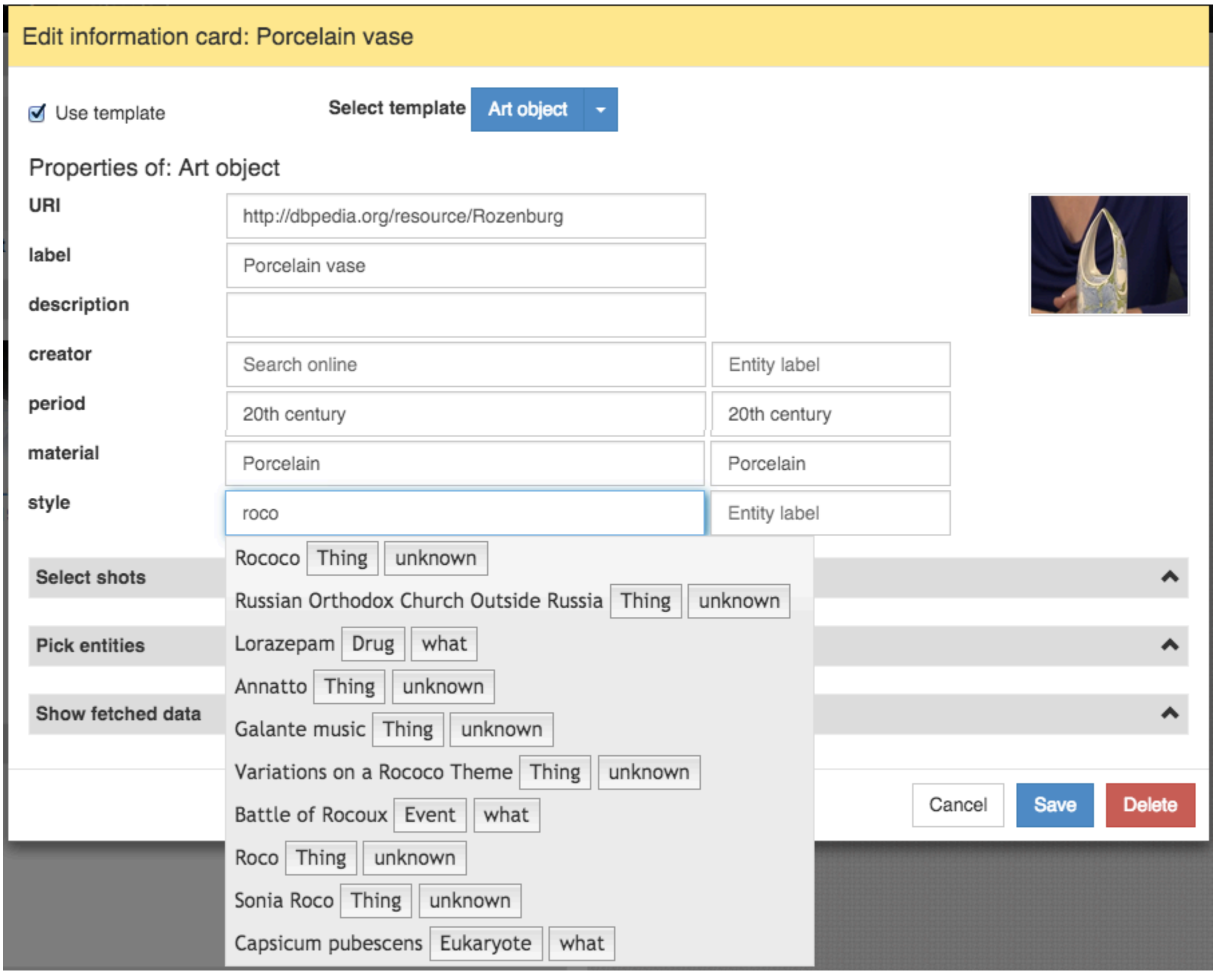
☑ Use template

Select template: Art object

Properties of:

label		
description		
creator	Search online	Entity label
period	Search online	Entity label
material	Search online	Entity label
style	Search online	Entity label

Figure 17: *Information card edit dialog – showing an empty art object template*



Edit information card: Porcelain vase

☑ Use template

Select template: Art object

Properties of: Art object

URI	http://dbpedia.org/resource/Rozenburg	
label	Porcelain vase	
description		
creator	Search online	Entity label
period	20th century	20th century
material	Porcelain	Porcelain
style	roco	Entity label

Rococo Thing unknown
Russian Orthodox Church Outside Russia Thing unknown
Lorazepam Drug what
Annatto Thing unknown
Galante music Thing unknown
Variations on a Rococo Theme Thing unknown
Battle of Rocoux Event what
Roco Thing unknown
Sonia Roco Thing unknown
Capsicum pubescens Eukaryote what
Select shots
Pick entities
Show fetched data
Cancel
Save
Delete

Figure 18: *Information card edit dialog – showing an information card being edited*

Figure 18 shows the information card dialog partly filled out and with all possible elements showing:

- **Edit form**: here the template can be selected and the form can be filled out
- **Shot selection panel**: here a shot can be selected to be used as poster image for the information card
- **Entity selection panel**: for inspecting the properties of automatically generated entities
- **Entity details panel**: showing the properties of the selected entity in the entity selection panel

Before explaining the functionalities of each of these parts in detail, the editor currently has two different approaches to choose from:

- use one of the preconfigured templates to fill out
- manually create an information card

Both approaches as well as the purpose of the aforementioned panels are described in the following two sections.

5.5.1 Using templates

Using a template is the most straightforward way of adding an information card, following the steps described below.

- The user fills in the pregenerated form
- The user selects a shot using the shot selection panel
- The user saves the card

For choosing a template, the user needs to make sure to select one of the configured templates from the pull down list as shown in Figure 17. When done so, automatically all of the key value pairs configured (see Section 3.2) for that template are displayed in a form.

For each configured literal, e.g., ‘label’ and ‘description’, the form provides a single text field. For each configured entity, e.g., ‘creator’ or ‘period’, the form displays a DBpedia search box (see Figure 18) and a separate text field to be able to edit the label of retrieved DBpedia entities.

Filling out the form

The text fields are just plain input fields where the user can enter free text. Using the DBpedia search boxes, the user must choose a DBpedia concept in order to properly fill out the field. After selection of an entity, the user is allowed to change the label of the retrieved entity; the underlying DBpedia URI is not displayed to the user.

Selecting a shot

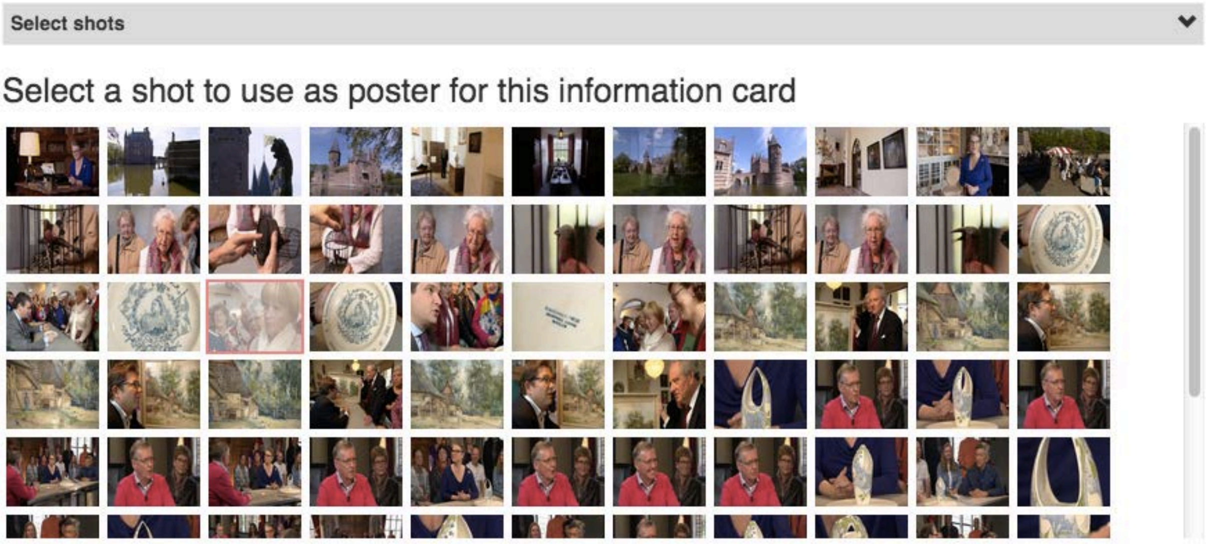


Figure 19: *Information card edit dialog - shot selection panel*

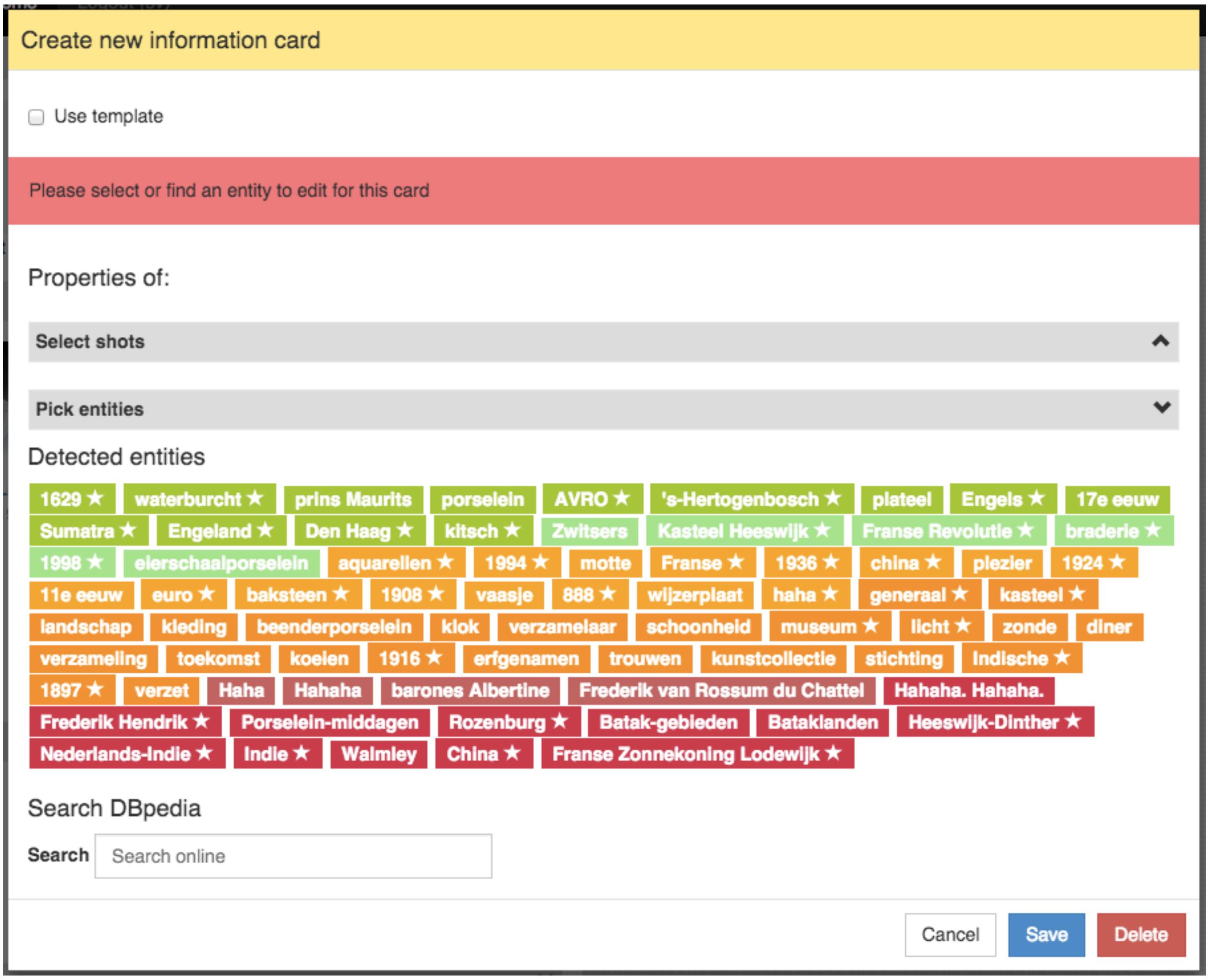
Using the shot selection panel as shown in Figure 19, the user can select one of the automatically detected shots to be used as a poster image for the information card. After filling in the form, and choosing a shot, the user can press ‘Save’ at the bottom of the dialog, to save the information card.

5.5.2 Creating custom information cards

If there is no suitable template configured, the user can add custom information cards based on either:

- an automatically detected entity
- an entity found using the DBpedia search box

For this, the user initially makes sure to deselect the “Use template” checkbox at the top of the dialog:



Create new information card

□ Use template

Please select or find an entity to edit for this card

Properties of:

Select shots

Pick entities

Detected entities

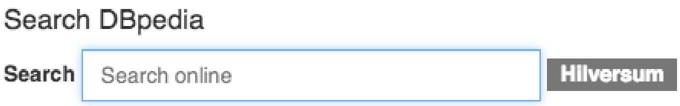
1629 ★ | waterburcht ★ | prins Maurits | porselein | AVRO ★ | 's-Hertogenbosch ★ | plateel | Engels ★ | 17e eeuw | Sumatra ★ | Engeland ★ | Den Haag ★ | kitsch ★ | Zwitsers | Kasteel Heeswijk ★ | Franse Revolutie ★ | braderie ★ | 1998 ★ | eierschaalporselein | aquarellen ★ | 1994 ★ | motte | Franse ★ | 1936 ★ | china ★ | plezier | 1924 ★ | 11e eeuw | euro ★ | baksteen ★ | 1908 ★ | vaasje | 888 ★ | wijzerplaat | haha ★ | generaal ★ | kasteel ★ | landschap | kleding | beenderporselein | klok | verzamelaar | schoonheid | museum ★ | licht ★ | zonde | diner | verzameling | toekomst | koelen | 1916 ★ | erfgenamen | trouwen | kunstcollectie | stichting | Indische ★ | 1897 ★ | verzet | Haha | Hahaha | barones Albertine | Frederik van Rossum du Chattel | Hahaha. Hahaha. | Frederik Hendrik ★ | Porselein-middagen | Rozenburg ★ | Batak-gebieden | Bataklanden | Heeswijk-Dinther ★ | Nederlands-Indie ★ | Indie ★ | Walmley | China ★ | Franse Zonnekoning Lodewijk ★

Search DBpedia

Search: Search online

Figure 20: *Information card edit dialog – creating a custom information card*

Next, the user can select an entity by either selecting one from the "Detected entities" list or search for one using the DBpedia search box.



Search DBpedia

Search: Search online

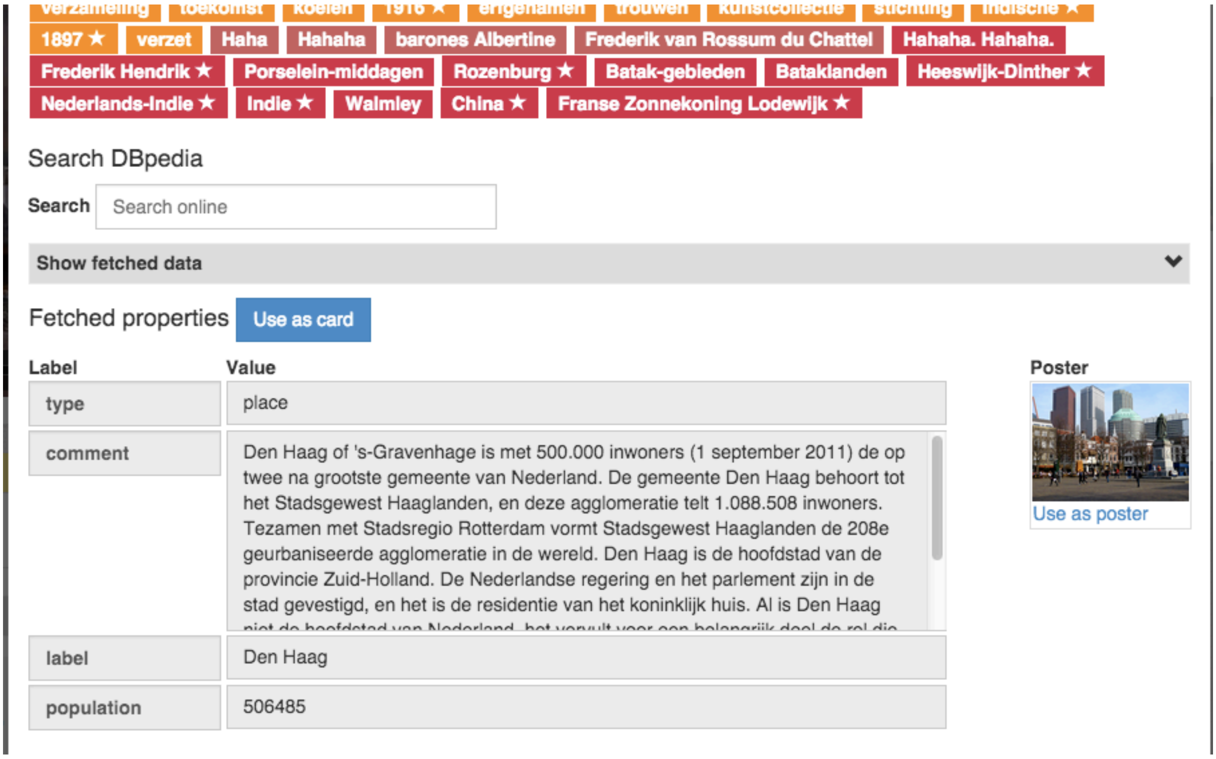
Hilversum

Figure 21: *Information card panel – the entity 'Hilversum' was found using the DBpedia search box*

When a user selects an entity from either the list (see Figure 20) or the one displayed after using the DBpedia search box (see Figure 21), CWI's EntityProxy (see Section 4.1.2) is called and fetches related properties by analyzing (if any) the entity's related DBpedia URL[28].

[28] The EntityProxy also accepts Wikipedia links, which are transformed into DBpedia URLs. This transformation however does not always work, as Wikipedia links do not always have direct DBpedia counterparts.

When successful the following information appears at the bottom of the screen (“Fetched properties”).



verzameling toekomst koelen 1916 ★ erfgenamen trouwen kunstcollectie stichting Indische ★
1897 ★ verzet Haha Hahaha barones Albertine Frederik van Rossum du Chattel Hahaha. Hahaha.
Frederik Hendrik ★ Porselein-middagen Rozenburg ★ Batak-gebieden Bataklanden Heeswijk-Dinther ★
Nederlands-Indie ★ Indie ★ Walmley China ★ Franse Zonnekoning Lodewijk ★

Search DBpedia

Search: Search online

Show fetched data

Fetched properties [Use as card]

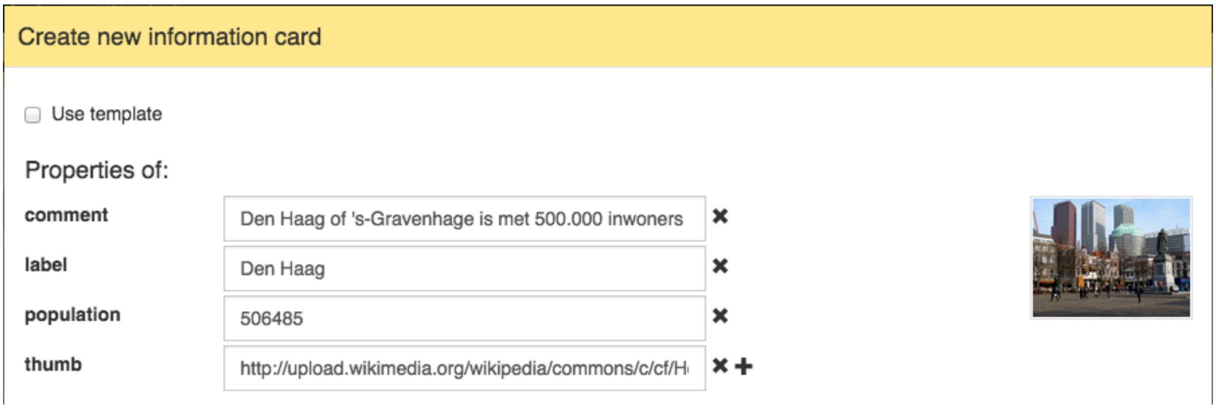
Label	Value
type	place
comment	Den Haag of 's-Gravenhage is met 500.000 inwoners (1 september 2011) de op twee na grootste gemeente van Nederland. De gemeente Den Haag behoort tot het Stadsgewest Haaglanden, en deze agglomeratie telt 1.088.508 inwoners. Tezamen met Stadsregio Rotterdam vormt Stadsgewest Haaglanden de 208e geurbaniseerde agglomeratie in de wereld. Den Haag is de hoofdstad van de provincie Zuid-Holland. De Nederlandse regering en het parlement zijn in de stad gevestigd, en het is de residentie van het koninklijk huis. Al is Den Haag niet de hoofdstad van Nederland, het vervult voor een belangrijk deel de rol die
label	Den Haag
population	506485

Poster

Use as poster

Figure 22: *Information card edit dialog - showing the data fetched from the EntityProxy*

Using the displayed information the user can decide whether the selected entity is indeed the one needed for this information card. If so the displayed properties can be attached to the card by clicking on the “Use as card” button. After this the “Fetched properties” are used to populate the form, representing the properties of the information card, at the top of the screen.



Create new information card

□ Use template

Properties of:

comment	Den Haag of 's-Gravenhage is met 500.000 inwoners ✖
label	Den Haag ✖
population	506485 ✖
thumb	http://upload.wikimedia.org/wikipedia/commons/c/cf/H ✖ ✚

Figure 23: *Information card edit dialog – after setting an entity as the basis for the information card*

Finally the user can edit the copied information to make sure it is appropriate for the eventual viewers of the program. In case the user does not appreciate the poster image copied from the selected entity, it is possible to select one by choosing a shot from the shot selection panel in the same manner as can be done when using the template (see Figure 19). After finishing the editing the user can save by clicking the save button, after which, the dialog is closed and the user is returned to the main screen.

5.6 Adding enrichments on demand

In the ET v1.0, the user could only select automatically detected enrichments from a static precalculated list (see D1.3). Moreover each of these enrichments was found based on a single automatically generated entity each, often being irrelevant or too unspecific.

After having tested this version of the ET with the TTK and RBB program teams the outcome was that they all thought that going through long lists of enrichments simply is too time consuming. Instead it was suggested that it would be very useful to be able to pick one or more automatically detected entities and use these as input for fetching enrichments *on demand*. This way the problem of solely having enrichments based on single entities is solved. Moreover, since the user requests the enrichments on demand, it is instantly clear on what basis each enrichment is found.

To support this new *on demand* approach EURECOM had to work on extending their enrichment tools, namely the TVEnricher and the TVNewsEnricher (see Section 4.1.1), with APIs that could be accessed by the ET in the following way:

- **Input**: a query
- **Output**: a set of hyperlinks with the following properties
    - **Source**: URL or name describing the source (website or system)
    - **Label**: describing the hyperlink
    - **Poster**: (if available) a relevant thumbnail

Unlike dimensions configured with the information card functionality (see previous section), dimensions configured with services supporting the *on demand* approach all use two dialog screens to fetch and edit enrichments:

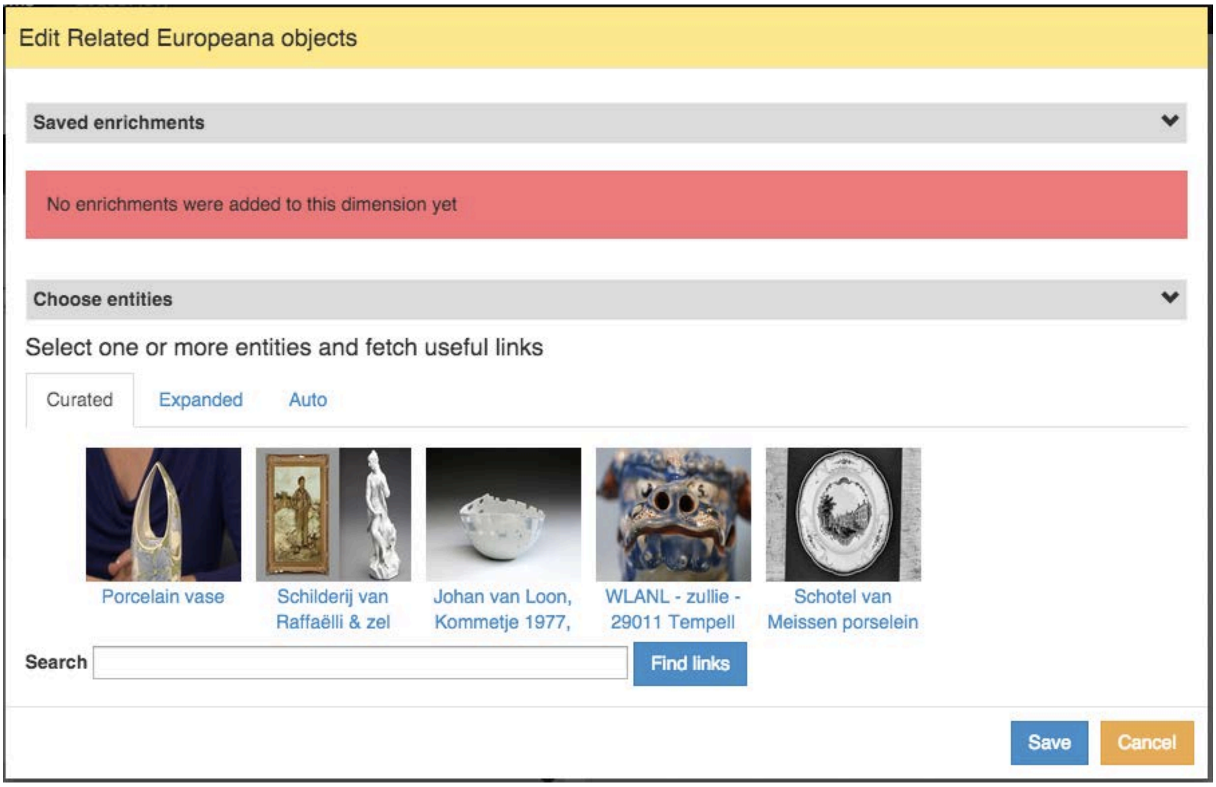
- searching enrichments on demand; adding/deleting multiple enrichments
- editing/deleting single enrichments

The reason for this difference in respect to the information card dialog is that for the on demand approach, many enrichments can be found and added at the same time, resulting in a different interaction design.

The detailed functionalities of the aforementioned dialogs are described in the following two sections.

5.6.1 Searching and adding new enrichments

When no enrichments have been added yet, the user can double-click on the ‘Search / Add’ (see Figure 16) button to open the search enrichment dialog (see an example in Figure 24 below).



Edit Related Europeana objects

No enrichments were added to this dimension yet

Select one or more entities and fetch useful links

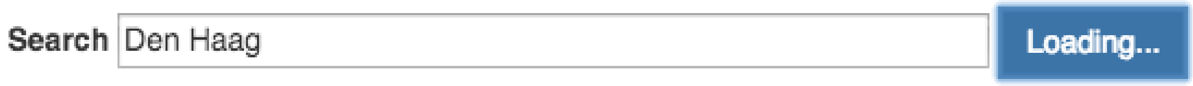
Saved enrichments
Choose entities
Curated
Expanded
Auto
Porcelain vase
Schilderij van Raffaëlli & zel
Johan van Loon, Kommetje 1977,
WLANL - zullie - 29011 Tempell
Schotel van Meissen porselein
Search
Find links
Save
Cancel

Figure 24: *Search enrichment dialog – before searching for enrichments*

The idea behind this dialog is the possibility to use either of the following to find related enrichments:

- **Curated enrichments**: these are all the information cards and enrichments that are already saved in the selected chapter
- **Expanded entities**: (only RBB news) these are entities that were fetched on demand using the EntityExpansion service. However, this has not been implemented yet.
- **Autogenerated entities**: these are the automatically detected entities

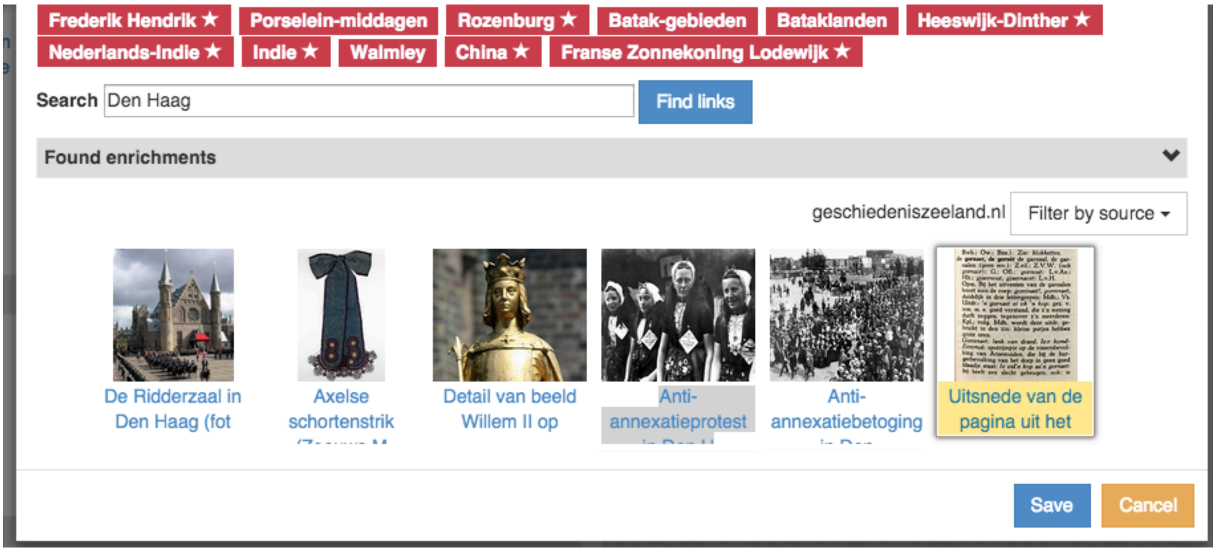
Each of these three sources can be inspected using the tabs displayed below “Choose entities”. If the user does not find any useful entities, it is always possible to enter search keywords in the search field at the bottom (see Figure 25). After choosing entities or entering a search query, the user can trigger the fetching of enrichments by pressing ‘Find links’. While searching the text of this button will change into ‘Loading…’



Search: Den Haag

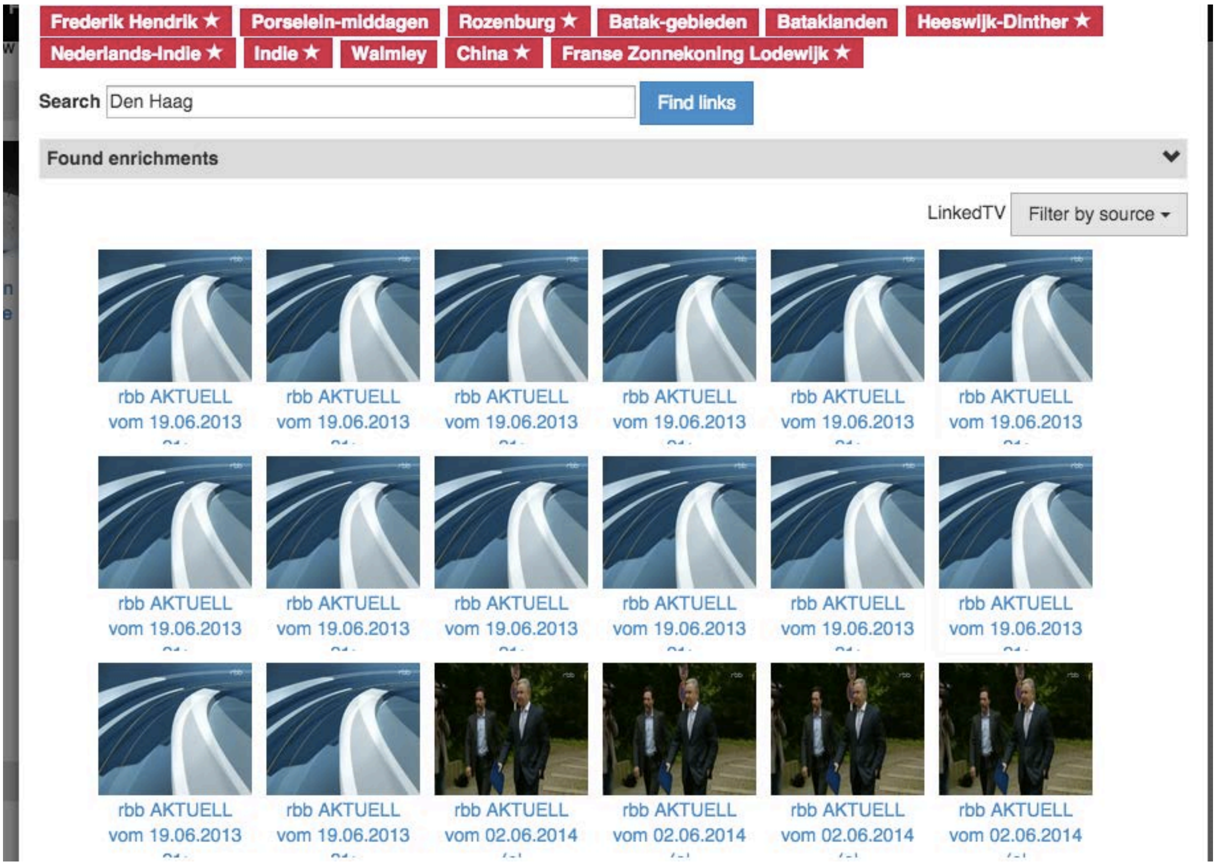
Figure 25: *Search enrichment dialog – while searching for enrichments*

After searching is done, the enrichments are displayed at the bottom of the page, as illustrated in Figures 26 and 27 below.



Search: Den Haag

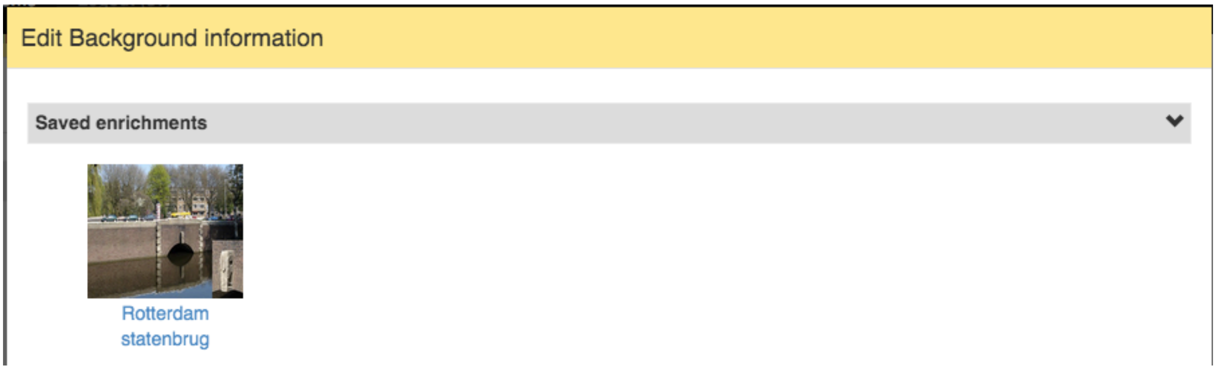
Figure 26: *Search enrichment dialog – after finding enrichments in the 'background information dimension'*


Frederik Hendrik ★
Porselein-middagen
Rozenburg ★
Batak-gebieden
Bataklanden
Heeswijk-Dinther ★
Nederlands-Indie ★
Indie ★
Walmley
China ★
Franse Zonnekoning Lodewijk ★
Find links
Found enrichments
LinkedTV
Filter by source
rbb AKTUELL vom 19.06.2013
rbb AKTUELL vom 19.06.2013
rbb AKTUELL vom 19.06.2013
rbb AKTUELL vom 19.06.2013
rbb AKTUELL vom 19.06.2013
rbb AKTUELL vom 19.06.2013
rbb AKTUELL vom 19.06.2013
rbb AKTUELL vom 19.06.2013
rbb AKTUELL vom 19.06.2013
rbb AKTUELL vom 19.06.2013
rbb AKTUELL vom 19.06.2013
rbb AKTUELL vom 19.06.2013
rbb AKTUELL vom 19.06.2013
rbb AKTUELL vom 19.06.2013
rbb AKTUELL vom 02.06.2014
rbb AKTUELL vom 02.06.2014
rbb AKTUELL vom 02.06.2014
rbb AKTUELL vom 02.06.2014

Search: Den Haag

Figure 27: *Search enrichment dialog – after finding enrichments in the 'related fragments' dimension*

Depending on which dimension, the results show different information[29], but can all be added and filtered in the same way. By simply double-clicking on one of the results it is added to the list of saved enrichments at the top (see Figure 28):


Edit Background information

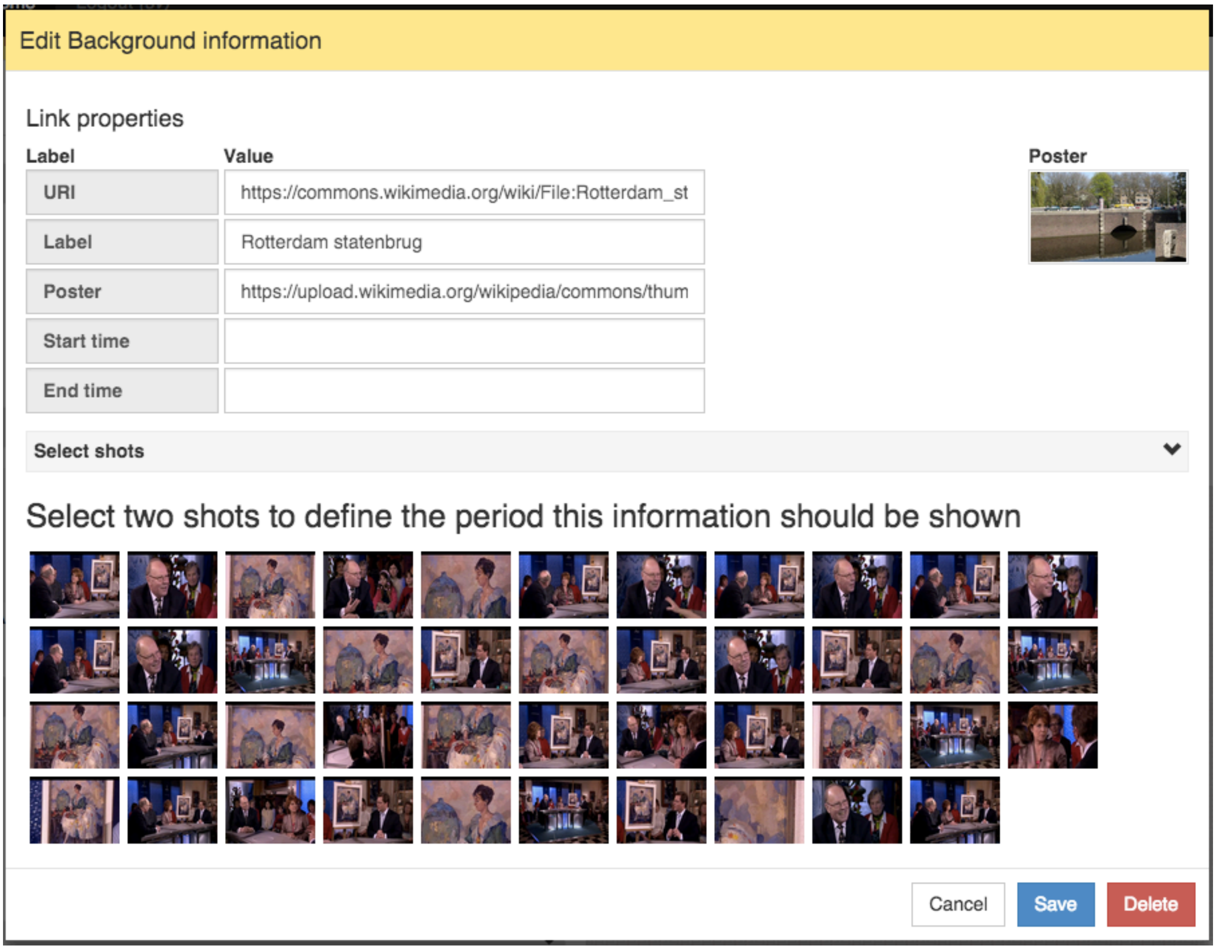
Saved enrichments

Figure 28: *Search enrichment dialog – after selecting an enrichment from the found enrichments.*

When selecting one or more enrichments in this way and subsequently pressing the 'Save' button at the bottom of the dialog, the enrichments are saved to the selected dimension within the currently active chapter.

Since this dialog is already crowded with different user interface elements, the enrichment can be further edited in a separate dialog, which can be opened by double clicking on a saved enrichment as shown in Figure 16. The dialog can be used as follows:

[29] The TVEnricher and TVNewsEnricher themselves support multiple dimensions



Edit Background information

Link properties

Label	Value
URI	https://commons.wikimedia.org/wiki/File:Rotterdam_st
Label	Rotterdam statenbrug
Poster	https://upload.wikimedia.org/wikipedia/commons/thum
Start time	
End time	

Poster
Select shots
Select two shots to define the period this information should be shown
Cancel
Save
Delete

Figure 29: *Enrichment edit dialog – editing a newly saved enrichment*

Next to editing the three most basic properties of an enrichment, namely the URI, label and poster, a user can also determine the start and end time when this enrichment should be displayed within the active chapter. Similar to the chapter edit dialog (see Figure 12), the (rough) boundaries of the selection can be conveniently set by choosing two shots from the shot selection panel.

Even though none of the LinkedTV players so far utilizes this information, this functionality adds much to the reusability of the ET as it can support use cases where editors need to be able to control the exact moment a piece of information is offered to viewers.

Lastly, using the enrichment edit dialog an enrichment can be deleted (from a dimension).

6 Customizing the ET

The current implementation of the ET is geared towards facilitating a usable platform for the current end users, namely RBB and TKK. In its design and architecture however, although not perfectly implemented as such yet, the ET is setup to enable the flexibility of extending and configuring a platform for various possible television programs and specific players working with LinkedTV technology.

The following sub-sections describe the different aspects of what potentially make the ET a reusable toolkit for configuring and implementing program specific editor tools. To avoid going into implementation details and since not everything has been implemented yet, the following will mostly be on a conceptual level.

6.1 Open source

The basic prerequisite for being able to produce a tool that can be used by many different users is to make it open source. Nowadays GitHub[30] seems the most popular platform and therefore the ET has been published there as well.

By advertising and offering technical support, LinkedTV aims to attract contributors, which could both improve the code base and strengthen the relation with potential LinkedTV clients. Moreover by offering the flexibility to adapt (see Section 6.3) the ET for using non-LinkedTV services as well, there is a possibility to attract a wider audience of contributors, which can also further improve the ET for LinkedTV and non-LinkedTV interested parties alike.

6.2 Configuration

Next to being open source, the currently implemented configuration options offer interested parties by default the option of configuring the following:

- **Chapter dimensions**: in other words, which of LinkedTV's enrichment services should be used for what dimension.
- **Information card templates**: for particular programs, editors might need the possibility to provide viewers with reoccurring pieces of information. In such cases it is convenient for editors to configure templates for this information (see 5.5.1 and 3.2 for more details).

The following sub-section describes the details of how the dimension configuration works and how developers could extend the ET with new services or interface components to extend the number of possible configurable dimensions.

[30] https://github.com/beeldengeluid/linkedtv-editortool

6.3 Extending code – adding dimensions

Besides being able to completely change the ET code or using the currently available configuration to fulfill end user specific requirements, the architecture of the ET also incorporates a plugin structure for extending and/or customizing the following two aspects:

- Dimension service
- Dimension user interface dialog component

Currently, the ET v2.0 has two different “dimension services” implemented, namely the TVEnricher and the TVNewsEnricher, which can be configured as was shown in Figure 1[31]. Figure 30 shows how the dimension configuration is handled in the architecture.

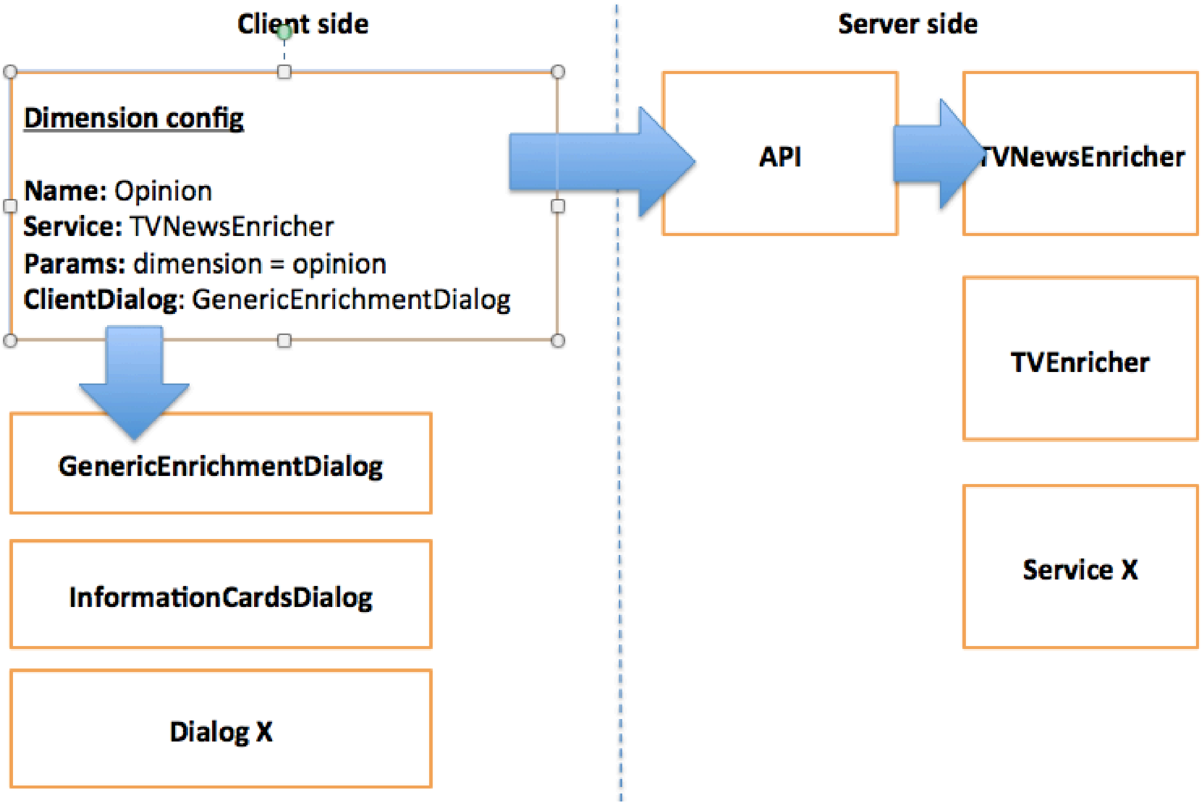


Figure 30: Architecture – how the ET handles the dimension configuration

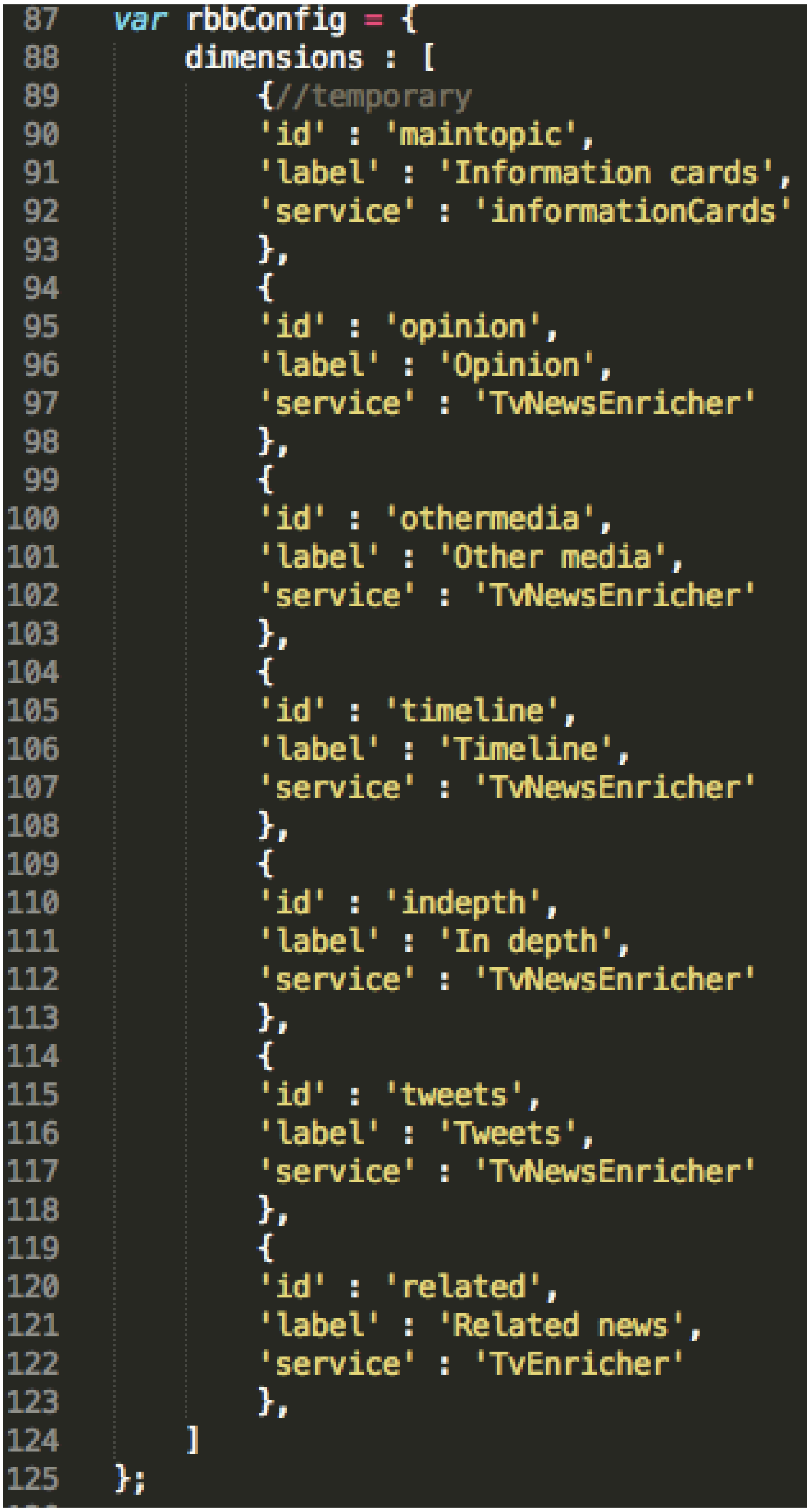
This figure presents an example configuration for a chapter dimension called ‘Opinion’ which uses the TVEnricher service with the parameters “dimension = opinion”. Moreover the

[31] Another available “service”, i.e. the “InformationCard”, is not really a service accessed via the ET API, but is rather an ingrained functionality of the ET. To keep the dimension configuration consistent, however, this option is configured via the “service” parameter.

parameter ClientDialog[32], i.e. the aforementioned user interface dialog component, is set to "GenericEnrichmentDialog", which will be explained further on.

First it is important to understand that each dimension service can support multiple dimensions. The most straightforward example is the TVNewsEnricher that supports 5 dimensions, namely: "in-depth", "opinion", "timeline", "other media" and "tweets", meaning that this service can be used for 5 different chapter dimensions. In Figure 30 the dimension "opinion" is configured via the "params" setting. For additional illustration, Figure 31 shows the RBB dimension configuration, which incorporates all of these 5 dimensions.

[32] This parameter has not been implemented yet



```
var rbbConfig = {
    dimensions : [
        {//temporary
        'id' : 'maintopic',
        'label' : 'Information cards',
        'service' : 'informationCards'
        },
        {
        'id' : 'opinion',
        'label' : 'Opinion',
        'service' : 'TvNewsEnricher'
        },
        {
        'id' : 'othermedia',
        'label' : 'Other media',
        'service' : 'TvNewsEnricher'
        },
        {
        'id' : 'timeline',
        'label' : 'Timeline',
        'service' : 'TvNewsEnricher'
        },
        {
        'id' : 'indepth',
        'label' : 'In depth',
        'service' : 'TvNewsEnricher'
        },
        {
        'id' : 'tweets',
        'label' : 'Tweets',
        'service' : 'TvNewsEnricher'
        },
        {
        'id' : 'related',
        'label' : 'Related news',
        'service' : 'TvEnricher'
        },
    ]
};
```

Figure 31: *JavaScript configuration of the RBB dimensions*

Getting back to Figure 30, when a contributor to the ET is interested in implementing a different dimension service, e.g., an API interfacing with a particular program database, this can be done by adding a component to the ET back-end that communicates to this API and translates its output to be compatible with the LinkedTV model[33].

Finally, when eventually implemented, using the “ClientDialog” option, a developer can point to a custom JavaScript module that takes care of showing an editing dialog in the same manner as e.g. the dialog that was already implemented for fetching multiple enrichments (see e.g., Figure 24). This last option offers developers a relatively flexible and convenient way of adding interface components specifically designed for the end-user.

6.4 Extending code – outlook

After presenting several means to adapt the ET to suit the specific needs of particular end-users, there are several points on which the ET can be improved as a valuable toolset for dynamically fetching enrichments and annotating video using these enrichments. To name a few:

- different configurable panels for editing chapters, including e.g. a timeline view
- overall timeline view showing chapters and annotations
- different configurable entity search boxes, instead of just the DBpedia search box (see Figure 21)

[33] The exact details of this can be found in the ET code.

7 Conclusions & future work

This document reports on the work done related to the development and release of v2.0 of the LinkedTV Editor Tool or ET. Initially, the work done for involving the end users at RBB and AVROTROS, as well as the result of this work, i.e., the updated user requirements were described in detail. Most notably, these new requirements include the major distinctive feature of the ET v2.0, namely the ability to fetch enrichments *on demand*. The actual implementation of this feature was achieved as a result of close collaboration with EURECOM who extended their TVEnricher, TVNewsEnricher and EntityExpansion web services to support on demand requests.

Following this, two newly conceived concepts, namely the information cards and chapter dimensions were introduced. Both concepts emerged from user tests, conducted by CWI, on the *Linked News Demonstrator* and showed potential in increasing the extensibility of the ET for different use cases. Specifically: information cards allow editors to have more control on exactly what information they approve of showing in the player. Subsequently, chapter dimensions enable selected enrichments, including information cards, to be grouped (by dimension) possibly narrowing the gap between what editors see in the ET and what is shown in a LinkedTV player.

Based on the updated user requirements and the newly introduced concepts, the system architecture was updated. This work included updating the back-end with new functionalities for interacting with the aforementioned EURECOM services as well as translating the output of these services to the ET data model. Moreover the back-end was extended to include communication with the EntityProxy developed by CWI. In order to adapt to the new data model and to increase the convenience for possible contributors of the ET's open source base in GitHub, the front-end of the ET was migrated from Backbone.js to AngularJS.

After describing the context and rationale of everything new in v2.0, the main body of this document described the resulting functionalities of the ET, which include 1) the new home page, 2) login mechanism, 3) selection of videos 4) curation of chapters, 5) addition of information cards and finally 6) the addition of other enrichments

Finally for future purposes this report included a special section on the customizability and extensibility of the ET with the aim of demonstrating its potential for supporting future use cases as well as its aims to become a fully-fledged developer toolkit, directly contributing to LinkedTV's library of products.

7.1 Future work

Following the efforts of publishing of the ET code in somewhat reusable form in GitHub, the future work will mostly entail improving the reusability and extensibility of the code in order to offer interested parties a developer toolkit with which it is possible to install and build editor tools that are tailored for specific program editing teams. To this direction the Netherlands Institute for Sound and Vision already expressed interest in further developing the ET for different projects. All further improvements from NISV will be directly updated in the current ET GitHub project.

Finally for LinkedTV's final round of evaluations, the ET is also planned to be evaluated by end users. Besides evaluating the usability of the tool itself, the ET will also most likely be used to evaluate the quality of the shot and chapter segmentation produced in WP1.

8 Bibliography

D1.3: *LinkedTV annotation tool, first release*	Jaap Blom (S&V); Nicolas Patz (RBB)
D1.4: *Visual, text and audio information analysis for hypervideo, final release*	Evlampios Apostolidis, Nikolaos Gkalelis, Fotini Markatopoulou, Damianos Galanopoulos, Eftichia Mavridaki, Christina Papagiannopoulou, Vasileios Mezaris (CERTH/ITI); Daniel Stein (FRAUNHOFER); Milan Dojchinovski, Tomáš Kliegr, Milan Šimůnek, Vaclav Zeman (UEP)
D2.2: *Specification of lightweight metadata models for multimedia annotation*	José Luis Redonda Garcia, Raphaël Troncy, Miroslav Vacura (EURECOM)
D2.6: *LinkedTV Framework for Generating Video Enrichments with Annotations*	Tomáš Kliegr, Jan Bouchner, Barbora Červenková, Milan Dojchinovski, Jaroslav Kuchař (UEP); José Luis Redonda Garcia, Raphaël Troncy (EURECOM); Jan Thomsen (CONDAT); Dorothea Tsatsou, Georgios Lazaridis, Pantelis Ieronimakis, Vasileios Mezaris (CERTH/ITI)
D3.5: *Requirements Document LinkedTV User Interfaces (Version 2)*	Lilia Perez Romero, Lynda Hardman, Michiel Hildebrand (CWI)
D3.7: *LinkedTV User Interfaces Selected and Refined*	Michiel Hildebrand, Lilia Perez Romero (CWI); Daniel Ockeloen, Pieter van Leeuwen (Noterik)
D6.3: *User trial results*	Katarina Stanoevska, Frederic Junker (University of St. Gallen); Mar Rodriguez (RBB); Lotte Belice Baltussen, Evelien Wolda (The Netherlands Institute for Sound and Vision); Mieke Leyssen (CWI)

9 Glossary

Annotation	A video annotation. Any piece of information/metadata assigned to a certain part of a video or in other terms: **media fragment**.
Chapter	A **segment** of a video encompassing a certain topic or scene (e.g. such as a film scene) or item (e.g. in a news program)
Chapter dimension	An annotation layer that has specific semantics defined in line with its intended use for a related LinkedTV player.
Entity	(Automatically detected) **named entity**
Information card	A construct that allows program editors to conveniently curate information, possibly derived from autogenerated **entities**, that is to be shown to viewers
Named entity	As defined in http://en.wikipedia.org/wiki/Named-entity_recognition#Named_entity_types
Shot	A **segment** of video composed by sequential frames captured uninterruptedly from a single camera.
Segment	A part of a video. Segments in LinkedTV are expressed as an **annotation**. Currently there are two types of segmentations, namely **shot**- and **chapter** segmentations.