D7.5: Functional specifications for social semantic functions (incl. prototype description)

<table>
<thead>
<tr>
<th>Revision</th>
<th>Draft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date of submission</td>
<td>31.03.2013</td>
</tr>
</tbody>
</table>
| Author(s) | Marlies Olensky, Humboldt-Universität zu Berlin  
Maarten Brinkerink, Beeld en Geluid  
Pieter Vijn, Beeld en Geluid  
Danny Sedney, Beeld en Geluid  
Antoine Isaac, Europeana |
| Dissemination Level | Public |
### Revision History

<table>
<thead>
<tr>
<th>Revision No.</th>
<th>Date</th>
<th>Author</th>
<th>Organisation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draft 1</td>
<td>17.01.2013</td>
<td>Marlies Olensky</td>
<td>Humboldt-Universität zu Berlin</td>
<td>First draft version</td>
</tr>
<tr>
<td>Draft 2</td>
<td>28.02.2013</td>
<td>Marlies Olensky</td>
<td>Humboldt-Universität zu Berlin</td>
<td>With input from Maarten Brinkerink, Juliane Stiller and Antoine Isaac</td>
</tr>
<tr>
<td>Draft 3</td>
<td>04.03.2013</td>
<td>Marlies Olensky</td>
<td>Humboldt-Universität zu Berlin</td>
<td>With input from Maarten Brinkerink, Juliane Stiller and Antoine Isaac</td>
</tr>
<tr>
<td>Draft 4</td>
<td>05.03.2013</td>
<td>Marlies Olensky</td>
<td>Humboldt-Universität zu Berlin</td>
<td>With input from Maarten Brinkerink, Juliane Stiller and Antoine Isaac</td>
</tr>
<tr>
<td>Draft 5</td>
<td>14.03.2013</td>
<td>Marlies Olensky</td>
<td>Humboldt-Universität zu Berlin</td>
<td>With PyBossa</td>
</tr>
<tr>
<td>Draft 6</td>
<td>25.03.2013</td>
<td>Marlies Olensky</td>
<td>Humboldt-Universität zu Berlin</td>
<td>With feedback from Maarten Brinkerink and Antoine Isaac</td>
</tr>
<tr>
<td>Version 1</td>
<td>28.03.2013</td>
<td>Maarten Brinkerink, Danny Sedney, Pieter Vijn, Marlies Olensky</td>
<td></td>
<td>Addition of prototype description</td>
</tr>
</tbody>
</table>

### Statement of originality:

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.
Contents

Contents ........................................................................................................................................... 3
1. Scope of this document ................................................................................................................. 4
2. Use cases ...................................................................................................................................... 5
  2.1. Waisda? ................................................................................................................................. 5
  2.2. PyBossa .................................................................................................................................. 9
3. Prototype ..................................................................................................................................... 15
  3.1. Waisda? for Europeana .......................................................................................................... 15
  3.2. PyBossa for Europeana ........................................................................................................... 21
4. References ................................................................................................................................. 26
5. Annex ......................................................................................................................................... 28
1. **Scope of this document**

The scope of this document is to provide functional specifications for social semantic functions in Europeana. These specifications complement the work being accomplished in Task 7.2, where innovative applications for user interaction and User Generated Content (UGC), in particular User Generated Meta Data (UGMD), will be developed, as it adds a semantic component to these applications. Based on a review of and inspiration from existing approaches (Carminati, Ferrari, & Perego, 2012; Choudhury, Breslin, & Passant, 2009; Garcia-Castro, Labarga, Garcia, et al., 2010; Kim, Scerri, Passant, et al., 2011; Marshall, 2009; van Hooland, 2006; Von Ahn & Dabbish, 2008; Zarro & Allen, 2010), this document specifies the functionality that will be implemented in the applications Waisda? and PyBossa. For both applications prototype code will be delivered and is described in the last section of the document.

For the remaining project phase, a process of feeding back the user-generated metadata into Europeana will be dealt with as well as the extension of the prototype code with more SKOS vocabularies. Also, the representation of UGMD in EDM will be tackled. One way of accomplishing this that will be investigated is using the Open Annotation Model (Hunter et al., 2010; Haslhofer et al., 2011 & 2012).
2. Use cases

Voss (2007) based on Marlow et al. (2006) formulates questions that need to be answered in order to classify the typology of tagging systems and that help to express use cases for tagging systems. The questions have been considered for describing both systems (the list of questions can be found in the Annex).

2.1. Waisda?

Waisda? is a crowdsourcing video annotation software that has been released as an open source framework by Sound and Vision. Waisda? will be further developed mainly through choosing additional controlled vocabularies that allow for better contextualization of objects and experimenting with an “emotional” vocabulary. An instance of the application will be build with data from the Europeana API, which is a substantial infrastructural alteration. The idea for the future (after the project ends) is either for Europeana themselves to offer such a game application from their website or that contributing data providers make it available directly on their own website.

Generally, it was agreed to have the game interface translated from Dutch into English. Further multilingual aspects will be taken into account in the second prototyping / development phase (starting April 2013). The following description contains the current state, marked with the round black bulleting points, as well as the alterations, additions, etc. of the functional specifications, marked with the arrowed bulletin points.

The user browses to the website of the game. Every guest user can play the game; there is no need for registration. Yet, at the end of the game the user has the option to register in order to save his score.

➢ While this keeps a low entry barrier for the user, the game needs to encourage the registration more. Only registered users can be analysed and their trustworthiness of contributions can then be assessed in the future. On the game recap page (at the end of the game), there must be a clear link that takes the user to the registration page. A separate box prominently displayed should be added. (cf. Figure 1 depicting the recap page).
The current configuration of the landing page loads five random videos to play the game with, for each respective 'channel' (a channel is a subset of the entire database of available videos, for instance based on a certain series of episodes or videos from a certain provider). If there are already game sessions with players ready to start, a pop-up will enter the screen to notify the user of the possibility of joining a game with other users. The users are still free to ignore this pop-up and can also start their own game, based on one of the five channels.)

- No alteration foreseen. In Europeana, a channel is a collection.

The user clicks on a video to start playing (they can start playing anonymously).

- Cf. first comment.

The user is referred to a ‘waiting room’ with a 20 second start-up time, that allows the game session to attract additional players and/or load previous sessions as ‘ghost players’ (in this ‘waiting room’ players also get the basic game instructions).

- For the second development phase, a more efficient use of the waiting time, in particular for recurring users will be considered (e.g. fulfilling small tasks, the option to register).

The game starts and the video plays from start to end. The user can immediately start tagging in the entry field below the video. When a user hits the [ENTER] button, the tag entry is added to the database. Tag entries include temporal information related to the video, based on the moment the player starts typing and the moment a tag is entered.

- Each tag entry earns points for the players, according to the following schema:
  - Tag entry: 5
  - Match (with another user’s tag): 50
  - Introduction of a tag (once it is matched by another player): 100 bonus
  - Geographical name: 25 bonus
  - Person name: 25 bonus
D7.5: Functional specifications for social semantic functions

- If tag entries added by players match within a ‘window’ of 10 seconds, the players earn points. Following the Games with a Purpose principle as developed by Louis von Ahn (von Ahn & Dabbish, 2008), the assumption is that blind matches constitute relevant tags.
- Introducing a word to be matched for the first time for a video item, earns the player points.
- Thanks to the use of Cornetto (Dutch language version of Wordnet) in the background, players can also score points by matching with synonyms of words used by opponents.
- Players get bonus points for tag entries that are:
  - Geographical names (matched against the GTAA thesaurus geographical names axis)
  - Person names (matched against the GTAA thesaurus person names axis)

As semantic disambiguation is an important aspect of user-generated metadata, the system cannot rely on correctly matching the users’ tags to the correct concept in the vocabulary. This is especially true when a new concept is introduced for a video, as the system has no other context to rely on.

In the second development phase, we will implement the following functionality for semantic disambiguation:

On the game recap page, the user has the option to review tags that could not be matched unambiguously and then disambiguate the tags and “approve” automatic matches to the controlled vocabularies. 5-10 tags from opponent players will be presented to the user and the disambiguation will earn the user additional points (25 points per match analogously to the matching of a geographical or person name).

As part of the second development phase, we also contemplate to experiment with a trustworthiness score of users and their tags respectively. Ideas for this phase include: possible cooperation with the SEALINCMedia project, age of the account, consistency in playing the game, average number of matches related to all tags entered, etc. In this phase, infrastructural issues will be important as well, such as when to calculate the trustworthiness score, how to store it (for each annotation? What are the implications on the system’s performance?), how to export it, etc. For the moment, a basic ratio should be stored for each user, after each game: ratio of simple (non-matched) tags vs. matched tags.

Additionally (to the GTAA Thesaurus), the system should consider additional controlled vocabularies for checking users’ tags. The following table is based on the work from Work Package 2 in the EuropeanaConnect project and represents a list of controlled vocabularies that have been available in the “semantic data layer”:

<table>
<thead>
<tr>
<th>Vocabulary name</th>
<th>Lang.</th>
<th>Description/Type</th>
<th>Comments/status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pico</td>
<td>lt</td>
<td>Concepts</td>
<td></td>
</tr>
<tr>
<td>GTAA</td>
<td>nl</td>
<td>Concepts, Geo</td>
<td></td>
</tr>
<tr>
<td>SC Ran</td>
<td>en</td>
<td>Concepts</td>
<td></td>
</tr>
<tr>
<td>Lausanne musea thesaurus</td>
<td>fr</td>
<td>Concepts</td>
<td></td>
</tr>
<tr>
<td>Fondazione Zerri vocabularies</td>
<td>it</td>
<td>Concepts</td>
<td></td>
</tr>
<tr>
<td>SWD</td>
<td>de</td>
<td>Concepts</td>
<td></td>
</tr>
<tr>
<td>Cornetto</td>
<td>nl</td>
<td>Concepts</td>
<td></td>
</tr>
<tr>
<td>DISMARC</td>
<td>en</td>
<td>Concepts</td>
<td></td>
</tr>
<tr>
<td>Iconclass</td>
<td>de,en,fr</td>
<td>Concepts</td>
<td></td>
</tr>
<tr>
<td>LCSH</td>
<td>en</td>
<td>Concepts</td>
<td></td>
</tr>
<tr>
<td>AATNed</td>
<td>nl</td>
<td>Concepts</td>
<td></td>
</tr>
<tr>
<td>Unesco</td>
<td>en,es,fr</td>
<td>Concepts</td>
<td></td>
</tr>
</tbody>
</table>
Table 1. Vocabulary selection from EuropeanaConnect

<table>
<thead>
<tr>
<th>Vocabulary Name</th>
<th>Language</th>
<th>Concepts</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wordnet 2.0</td>
<td>en</td>
<td>Concepts</td>
<td></td>
</tr>
<tr>
<td>Geonames</td>
<td>mul</td>
<td>Geo</td>
<td></td>
</tr>
<tr>
<td>WOLF Wordnet</td>
<td>fr</td>
<td>Concepts</td>
<td></td>
</tr>
<tr>
<td>Wordnet 3.0</td>
<td>en</td>
<td>Concepts</td>
<td></td>
</tr>
<tr>
<td>Rameau</td>
<td>fr</td>
<td>Concepts, Geo</td>
<td></td>
</tr>
<tr>
<td>Austrian Mediathek thesaurus</td>
<td>de</td>
<td>Concepts</td>
<td></td>
</tr>
<tr>
<td>Amsterdam Museum Vocabularies</td>
<td>nl</td>
<td>Concepts, Geo, Persons</td>
<td></td>
</tr>
<tr>
<td>EUScreen</td>
<td>ca, da, de, en, el, hu, it, nl, sv</td>
<td>Concepts</td>
<td></td>
</tr>
<tr>
<td>Bibliopolis thesaurus</td>
<td>nl</td>
<td>Concepts</td>
<td></td>
</tr>
<tr>
<td>OSZK Thesaurus</td>
<td>hu</td>
<td>Concepts, Geo</td>
<td></td>
</tr>
<tr>
<td>Polish Subject headings (JHP BN)</td>
<td>pl</td>
<td>Concepts</td>
<td></td>
</tr>
<tr>
<td>GeonetPT</td>
<td>pt</td>
<td>Geo</td>
<td>For exploitation and sharing within the Europeana network</td>
</tr>
<tr>
<td>VIAF</td>
<td>mul</td>
<td>Persons</td>
<td></td>
</tr>
<tr>
<td>CSIC (Spanish subject headings)</td>
<td>es</td>
<td>Concepts</td>
<td>only for use in Europeana</td>
</tr>
<tr>
<td>Getty AAT</td>
<td>en</td>
<td>Concepts</td>
<td>only for research purposes, in Europeana</td>
</tr>
<tr>
<td>Getty ULAN</td>
<td>mul (labels are not language-tagged)</td>
<td>Persons</td>
<td>only for research purposes, in Europeana</td>
</tr>
<tr>
<td>Getty TGN</td>
<td>mul (but labels are not language-tagged)</td>
<td>only for research purposes, in Europeana</td>
<td></td>
</tr>
<tr>
<td>Joconde</td>
<td>Fr</td>
<td>Concepts, Geo, Persons</td>
<td>only for research purposes, in Europeana</td>
</tr>
</tbody>
</table>

Even though most of these vocabularies are quite specific, it will be still useful to use them in connection with specific collections. Therefore, the application will support the possibility to configure individual tasks (which would be done by the institution employing the game application). Specific collections (more than one) can be chosen to be part of the tagging game and according to those, appropriate controlled vocabularies can be chosen to be used for matching. For the prototype code, the Open Images collection will be chosen and the EU screen vocabulary will be used to match the tagged terms. In general, a generic tool will be developed that will allow to add collections and

---

1 Gligorov et al. (2011) find that 59% of the users’ tags could not be verified or found in a vocabulary. It is obvious, that more matches were found in the general Cornetto than in the GTAA. That is why, it is important to choose an “appropriate” vocabulary, corresponding to the collection.
vocabulary. For DBpedia, only the Wikipedia categories\(^2\) are available in SKOS\(^3\). Further discussion is needed whether DBpedia could still be used as well as other larger vocabularies that have been used for semantic enrichment of Europeana (Geonames and GEMET). As time period vocabulary, we will wait for the results from the Europeana 1914-18 project that will investigate a combination of DBpedia and LCSH. The methodology can then be re-used to extract further time periods from DBpedia that go beyond World War I. The prototype showcases the infrastructure for doing this with GTAA – the actual importing of SKOS vocabularies will be done as part of phase two.

Moreover, we will experiment with an “emotional” vocabulary. So, in addition to tagging the game with terms, in the instructions, the user will be encouraged also to tag the video with emoticons. These will be matched against an “emoticon vocabulary” and the points will be handled the same way as with the other controlled vocabularies. This is especially useful for the multilingual user. The use of this metadata for the search (in the Europeana interface or other applications) can be investigated separately. An emoticon vocabulary has been compiled consisting of a list of standardised emoticons from Wikipedia (http://en.wikipedia.org/wiki/List_of_emoticons#cite_note-cool-smiley-4), complemented by input from an Emotion Ontology\(^4\).

After the video is done playing, the user is redirected to a game recap page. This provides the user with a summary of the points that were gathered during the game session, how they performed in relation to the opponents during the game sessions and an overview of all tag entries that received points and the rationale behind this.

- Users that want to save their score and compete for a spot in the leader boards can register an account by providing the system with an e-mail address, password and user name (they also have to comply with the Terms of Use); they will also receive their own profile page with a game history and personal statistics.
- As mentioned before, the game recap page must have a clear link which encourages the user to register and save their score. Tags from recurring users with high scores could be treated differently in terms of trustworthiness of their contribution. Yet, this will not be part of the functional specifications or the prototype code but could be part of further research during the rest of the project.

Users are also able to share their score on several social media networks (the AddThis Plugin\(^5\) is used).
- No alteration foreseen.

### 2.2. PyBossa

PyBossa is a platform for creating and running crowd-sourcing applications that utilize online assistance in performing tasks that require human cognition, knowledge or intelligence such as image classification, transcription, geocoding and more (Brinkerink, 2012). For this task, we take the instance of PyBossa that was created for the Amsterdam Museum as a baseline.

\(^2\) http://downloads.dbpedia.org/preview.php?file=3.8_sl_en_sl_skos_categories_en.nt.bz2  
\(^3\) http://wiki.dbpedia.org/Downloads38#categories-skos  
\(^4\) http://bioportal.bioontology.org/ontologies/1666  
\(^5\) http://www.addthis.com/
and adapt it. In this sense, we understand and turn PyBossa into a generic platform for creating online tagging games for still images that lets users perform classification tasks. Analogously to Waisda?, an instance of the application will be build with data from the Europeana API. Also for this application, the idea for the future (after the project ends) is either for Europeana themselves to offer such a game application from their website or that contributing data providers can easily create their own instance of an online tagging game for still images, based on a selection of Europeana records. Europeana does not host the actual objects (in this case the still images) but only the metadata records. This requires that selected collections for an instance of the tagging game contain resolvable references to the actual online content, hosted by the data provider (Brinkerink, 2012).

Generally, it was agreed to have the game interface translated from Dutch into English. Further multilingual aspects will be taken into account in the second prototyping / development phase (starting April 2013). The following description contains the current state, marked with the round black bulleting points, as well as the alterations, additions, etc. of the functional specifications, marked with the arrowed bulletin points.

Editors (in this case the Amsterdam Museum) can create series and questions (tasks). In Figure 2 to the right, editors can select a collection set and preview the relevance of this set. To the left, the range of questions is presented and can be edited.

- create new questions (= information needs) in the CMS
- determine which tags are saved per task on basis of the responses.
- search the linked collection database to create new sets of the collection (= series)
- determine what questions within a set should be answered

![Figure 2. Editor Page](image)

- Editors should be able to compile tasks that consist of a collection in Europeana, then select appropriate vocabularies (that can be used to describe the objects) for it and select specific tasks (described in the user section): linking documents, correct metadata, correct enrichments. The selection of collections can be limited to a certain
data provider, but can also be a combination of several collections from different data providers or a thematic selection. The selected tasks can be combined to create a set of tasks – this functionality will be implemented in a later phase. The crux of making this work effectively is more about matching a collection with a certain ‘information need’ to a specific task that supports added value for this particular need. For each task editors can select one of the metadata profiles from the Europeana Portal (e.g. full, standard, search result profile) that will then be presented with the image(s) to the user. As part of the correcting metadata fields, the editor should have the possibility to choose specific metadata fields that will be “open for correction”.

In the admin section editors can perform different tasks that are related specifically to the instance of the Amsterdam Museum which limited the game to specific user groups. In general, the game should be open to any user. So, we will not need the restriction to be part of a school or any specific group. We will not need the possibility of limiting the words that are saved as tags; also grouping of users will not be necessary.

Users can:
- visit the homepage of the tagging game
- register
- log in
- view what % of the set (records) is tagged (desired)
- see who did the most tags (desired)
- can re-request their password
- after log in choose which series (created by editors) to tag (cf. Figure 3)

The above-mentioned “desired functionality” could be implemented for the prototype with a low priority. For a user to see the progress on his task should help keeping the drop-out-rate low. This could be implemented as a progress bar covering the % of how many tasks of a task set a user has already completed. Additionally, a scoreboard, as available for Waisda?, could be useful to stimulate competition and participation, but is not mandatory.
The user will be presented with a set of tasks for a specific collection as specified by the editor (cf. section on editor for details on what constitutes a set of tasks). For the prototype, this will be a flat list of one collection and the three different tasks. Later on, the user should be able to choose between different collection as well as tasks (the order of this choice still needs to be determined).

- The three different types of tasks are
  - Linking documents: PyBossa should use the information on the standard IR-similarity that the Europeana API transmits, to present two images to the user that are similar to each other. The user should also see some metadata – I would suggest the title of the object and the creator. The user will then need to judge the two images and be presented with the question underneath the two images “Are these two images similar to each other?” and the possibility to answer with “yes/no” (via a radio button); and then underneath it, it should ask “Can you specify what type of relationship these two images have?” and have a field where the user then can type in the specification of the type of relationship. Here, we can use the typology of clusters that was developed in the course of an experiment by Europeana and OCLC (Charles & Wang, 2012):
    - Same objects/duplicates
    - Part of another Cultural Heritage Object (CHO)
    - Views of the same CHO
    - Derivative work
    - Part of thematic cluster
    - Collections

In the second development phase, there could be experiments with more than two images, e.g. presenting the user with clusters of images to comment on. An example could be this image:

http://www.europeana.eu/portal/record/2023707/1024E4CE3FE5E4D9E99A5A27089A6C0D7DB49C49.html

At the bottom of the metadata “similar content” is displayed and for example this image could be displayed as the second one.

http://www.europeana.eu/portal/record/2023707/22958D7D4483DD4A5BB3217C5AB3D17F01FDF375.html

Then the user is asked if he agrees that these images are similar to each other. And the type of relationship would be either there is the same person depicted or in both pictures there are women or the picture is about hats.

- Correcting metadata: For the task of correcting metadata, the user will be presented with only one image at a time. The metadata fields he will see with the image should be determined by the editor compiling the task. So, selecting the metadata fields will be made an option in the admin section, when the editor selects the collection (re-using the metadata profiles). Only the metadata fields that were selected by the editor as “open to correction” will then have underneath or next to each field the question “Do you agree with the description of the image?” as well as the answer possibilities “yes” and “no” (also via a radio button). Then again, if the answer is no, the user should get the possibility to provide a correct tag. In the second development phase, experiments with OpenSKOS will be made to see if it could be used for the autocompletion functionality. If the answer is yes, we still should allow for additional tags and ask the user “Can you provide additional and/or more specific information?” and then provide the same box as if he would correct the metadata but store this one as enrichment. In the end, the question is how a quality control of these corrections and additional tags can be managed. A manual check by the editor could be implemented at a later stage as “meta-tasks” that will check user-generated metadata for accuracy. Again, similar to Waisda? this will be left for future research (taking the trustworthiness of the
D7.5: Functional specifications for social semantic functions

users into account). And we would need to consider (at least some of) these strategies for semantic and multilingual enrichments (Olensky, Stiller & Dröge, 2012) while matching tags to controlled vocabularies (Table 2).

<table>
<thead>
<tr>
<th>Level</th>
<th>Areas of concern</th>
<th>Strategic execution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metadata</td>
<td>Metadata quality</td>
<td>Quality score for metadata, no enrichments below the score, data cleaning process</td>
</tr>
<tr>
<td>Metadata</td>
<td>Structure of metadata</td>
<td>Data normalization e.g. surname forename, rules for syntax, validate fields against a schema, consistency check for field refinements</td>
</tr>
<tr>
<td>Vocabulary</td>
<td>Choice of vocabulary</td>
<td>Choose domain-specific vocabulary or a subset of a vocabulary, exclusion of parts of the vocabulary</td>
</tr>
<tr>
<td>Vocabulary</td>
<td>Scope of enrichment</td>
<td>Choose fields to be enriched with a specific vocabulary or even limit enrichment to subsets or specific collections</td>
</tr>
<tr>
<td>Workflow</td>
<td>Semantics</td>
<td>Disambiguate metadata values and use context</td>
</tr>
<tr>
<td>Workflow</td>
<td>Named entities</td>
<td>Apply automatic named entity recognition</td>
</tr>
<tr>
<td>Workflow</td>
<td>Cross-lingual ambiguities</td>
<td>Metadata records and enrichment term need to have the same language</td>
</tr>
<tr>
<td>Workflow</td>
<td>Weighting of enrichments</td>
<td>If multiple values in one metadata field are enriched, they should be weighted according to their relevance</td>
</tr>
<tr>
<td>Workflow</td>
<td>Matching rules</td>
<td>Use exact matches, include variants from the controlled vocabulary, rule on how to enrich multiple values in a field</td>
</tr>
<tr>
<td>Workflow</td>
<td>Quality assurance</td>
<td>Quality checks (automatically or manually) before enrichments “go live”</td>
</tr>
<tr>
<td>Workflow</td>
<td>Quality assessment</td>
<td>Assess the scope of the enrichments with regard to their occurrence in search results</td>
</tr>
</tbody>
</table>

Table 2. Framework of strategies for semantic and multilingual enrichments

Here an example could be this image: http://www.europeana.eu/portal/record/09405u/3A69262579AA18C34F2616C08AA8587922E954D7.html which holds the Creator “Root”. So basically the user would mark this as incorrect, he might have information on the correct creator; but in this case it is rather unlikely.

- Correcting enrichments: This task is very similar to the previous one on correcting metadata. Only here the user will be presented some basic metadata information, like title of the object, date and creator and then he will see the metadata field(s) that have been enriched as well as the actual enrichment. The same principle as for the correcting metadata should be applied asking for the agreement of the user. In case of disagreement, he should be able to type in his suggestion – here an auto-completion box with the terms from the respective vocabularies used for enrichment makes sense: GEMET Thesaurus for dc:subject, dc:type, dcterms:alternative; DBpedia (or
DBpedix\textsuperscript{6}) for dc:contributor, dc:creator; and GeoNames for dc:coverage, dcterms:spatial. Like Waisda?, we will use as Time Ontology something methodologically based on the results of the experiment from Europeana 1914-18 with DBpedia and LCSH. And the same considerations about the trustworthiness and the matching with the vocabularies need to be applied as described in the upper section on correcting metadata.

As an example I take one the wrong enrichments from our evaluation: http://www.europeana.eu/portal/record/07202/49279933A922E75938223A44C27767724BC8E2F9.html. This image was enriched with Adolphe Mouron Cassandre, although the creator is Cassandre Guggiari. So, it is the wrong person and the reason behind this is that the matching tool did not consider the structure of the name (first vs. last) and accepted a partly match as correct match. The user would then be able to mark the enrichment as wrong and through DBpedix (with the autocompletion) try to find the correct creator.

\textsuperscript{6} The use of DBpedix (http://europeanalabs.eu/browser/contrib/ait/trunk/dbpedix) would be worth experimenting with. DBpedix is a semantic Tag-Autocompletion component. It is a backend to help users disambiguate their tags using DBpedia as the underlying ontology (providing also the corresponding URI) and source of multilingual labels.
3. Prototype

In order to demonstrate and test the suggested alterations in this document, prototype code has been delivered that already includes part of the alterations suggested in this document and will take aboard and further investigate them as part of the further development of the innovative applications that is part of Task 7.2. This section describes the current state of both prototype instances (Waisda? and PyBossa). Both prototypes have been translated into English as described in the functional specifications.

3.1. Waisda? for Europeana

http://waisda.tuxic.nl:8080/ (prototype instance)
https://github.com/beeldengeluid/waisda/tree/master-0.0.2-SNAPSHOT (prototype code repository)

3.1.1. Europeana Video Collection Importer

To facilitate the Europeana Network in setting up an instance of the Waisda? Video Labeling game, a reusable Europeana Video Collection Importer was added to the codebase of Waisda?, to allow for importing video datasets via the Europeana API.

Create a admin user

To import Europeana data the user must log in with an admin enabled user first. To create an admin user, register a new user in Waisda? frontend and set corresponding User table record's admin flag to '1'.

Enter a search query in import page

When logged in, browse to URL http://waisda.host.name/europeanaimport/start. The screen shown provides the user with a search input box where the user can enter a Europeana search query. See image below:

The items Europeana finds based on this search query are imported into the Waisda? database.

After entering a search query and hitting the 'start' button, the import page feeds back the total number of items found (any type, not only VIDEO). See image below:
If the search query doesn't meet the expectation, another query can be entered and the import page shows the total number of items to be imported again.

If this number meets the expectation, hit the 'start' button again to actually start importing the items. See image below:

Note that a search will be batched into one or more subsearches, with each subsearch fetching at most a configured number of items (see Config and Setup).

**Search Query formats**

Please have a look at Europeana API documentation to learn more about query formats. Basically, Europeana can query most fields it stores. For example, to look for a set of items of a given data provider one has to use the following search query:

```
provider_aggregation_edm_dataProvider:[Name of collection]
```

Where [Name of collection] is the name of the dataset to import (for example: 'Open Beelden')
Import filtering
The importer applies a filter to perform a best-effort importing only valid videos. A video is considered valid if:
- imageUrl <= 1024 characters
- sourceUrl <= 1024 characters
- sourceUrl matches one of the 'accepted URL' expressions (see Config and Setup section)
- the video source has duration filled in correctly\(^7\)
- the video hasn't been imported yet. Videos that have been imported already will be updated instead

Note that video source URLs are preferred to be read from the edm:isShownBy field. If not available, the importer analyzes the edm:WebResources field for a valid video URL. If none of these fields provides a proper video URL, the item is skipped.

Import progress
When an import is running, the importer page shows a summary that indicates the progress and the import log. The progress indication shows:
- currently imported title
- current index number
- number of total items

The import log shows the result of each imported item using the log levels INFO, WARNING and ERROR. So it logs when an URL is too long, no duration filled in etc. See image below:

Beside the frontend logging, Waisda? also creates an import log file named europeanaimport.log.

Stop the importer
The importer can run only one process at a time. It does not allow to start more than 1 import (at least when deployed as a single active node). To stop a currently running import, hit the stop button. The import log will show that the stop command was issued.

\(^7\) Format according to: http://en.wikipedia.org/wiki/ISO_8601#Durations
3.1.2. Waisda? for Europeana Prototype Instance

To showcase the possibilities with the further developed Waisda? codebase, a publicly available prototype instance was set up and clearly branded as a Europeana related project outcome (Figure 4).

As suggested in this document the prototype instance has implemented a call-to-action reminding unanimous guest players to register an account, as part of the game recap. This link directs them to a registration page and allows them to save points they have already gathered during their guest session (Figure 5 and 6).
D7.5: Functional specifications for social semantic functions

Figure 5. Game recap with a call-to-action for registration
The current version of the prototype instance tries to match tag entries against two controlled-vocabularies, namely the GTAA person names and the GTAA geographical names. At a later stage this will be expanded with additional controlled vocabularies that can be added to the game using a generic importer for SKOS vocabularies. In the example below (Figure 7) the player scores points for using a term from GTAA person names (Salvador Dali) and a term from GTAA geographical names (Rotterdam). In the game recap (Figure 8) these matches with different vocabulary terms (called 'dictionary matches in the game') can be identified with their own icon (a puppet for the GTAA person names terms and a pin for the GTAA geographical names).
3.2. PyBossa for Europeana


prototype code repositories:
- Europeana client API: [https://github.com/mk270/europeana-search](https://github.com/mk270/europeana-search)
- PyBossa “upstream” features: [https://github.com/PyBossa/pybossa](https://github.com/PyBossa/pybossa)
- Bespoke PyBossa tools: [https://github.com/mk270/pybossa](https://github.com/mk270/pybossa)

3.2.1. PyBossa task importer and task presenter

The Europeana Search client API is also available as “europeana-search” from [http://pypi.python.org](http://pypi.python.org), and thus by “pip install europeana-search”; it is a dependency of the Bespoke PyBossa tools.

PyBossa upstream features include:
- Heroku deployment support
- Importer generalisation
- Two major hygiene fixes to make code intelligible
- Fixes to bugs in task generation templates

Bespoke PyBossa tools:
- Europeana API task importer
- Multi-image task presenter support
- Task presenters:
  - Metadata correction game
There are lots of branches of code in the mk270/pybossa fork of PyBossa; the ones relevant to this project are:

**culttag-redux** - this set of tools are specially made for the tagging game but of no interest for the users of the platform PyBossa, because they are too specific.

**culttag-deployable** - the tools above, plus some Heroku-specific configuration information

To understand the meaning of “task presenter” and “task importer”, see [https://github.com/PyBossa/pybossa/blob/master/doc/overview.rst](https://github.com/PyBossa/pybossa/blob/master/doc/overview.rst)

**Licensing:** the source is under a mixture of GNU Affero GPL v3.0 and Apache Software License v2.0 licences.

### 3.2.2. PyBossa tagging app walkthrough

**Sign up**

Go to [http://culttag6.herokuapp.com](http://culttag6.herokuapp.com)

Click “Sign In” (unless for some reason you already have an account that’s logged in)

Ignore the input boxes and click “Create a New Account”

Enter details and click “Create an account”

(This should log you in)
Create an app

Click "Create" from the navigation bar at the top
Fill in the form - note that “Short Name” should be lower case, no spaces; e.g., “myapp1”
Click “Create the application”; you should get the app control panel

Click “Edit the task presenter”
Go the Cultural Tagging Game box and click “Use the image template”
Ignore the HTML text and just click “Update [...] presenter”; you should be returned to the app control panel
Click “Import tasks”
Go to the Europeana box and click “Use the Europeana API”
D7.5: Functional specifications for social semantic functions

Add a search string, e.g., “rembrandt”
Add your API key (you can use: j5x8xiiJE)
Click “Import” once. Be prepared to wait a few seconds before the page refreshes
You should be taken to the app landing page

Try an app
From the app landing page reached above, click “Start contributing now”
D7.5: Functional specifications for social semantic functions

**Settings**

From the login menu (in the menu bar, most to the right) My Applications can be accessed.

![Application interface screenshot]

Here you find the app control panel as described under ‘Create an app’. In the left menu you can find important features as export tasks (metadata), an oversight of tasks and statistics.
4. References


5. Annex

Questions from Voss (2007) classifying the typology of tagging systems:

Tagging Rights
   Who is allowed to tag resources?
   Can any user tag any resource or are there restrictions?
   Are restrictions based on resources, tags, or users?
   Who decides on restrictions?
   Is there a distinction between tags by different types of users and resources?

Source of Resources
   Do users contribute resources and have resources been created or just supplied by users?
   Or do users tag resources that are already in the system?
   Who decides which resources are tagged?

Resource Representation
   What kind of resource is being tagged?
   How are resources presented while tagging (autopsi principle)?

Tagging Feedback
   How does the interface support tag entry?
   Do users see other tags assigned to the resource by other users or other resources tagged with the same tags?
   Does the system suggest tags and if so based on which algorithms?
   Does the system reject inappropriate tags?

Tag Aggregation
   Can a tag be assigned only once to a resource (set-model) or can the same tag be assigned multiple times (bag-model with aggregation)?

Vocabulary control
   Is there a restriction on which tags to use and which tags not to use?
   Are tags created while tagging or is management of the vocabulary a separated task?
   Who manages the vocabulary, how frequently is it updated, and how are changes recorded?

Vocabulary Connectivity
   Are tags connected with relations?
   Are relations associative (authority file), monohierarchical (classification or taxonomy), multihierarchical (thesaurus), or typed (ontology)?
   Where do the relations come from?
   Are relations limited to the common vocabulary (precoordination) or can they dynamically be used in tagging (postcoordination with syntactic indexing)?

Resource Connectivity
   How are resources connected to each other with links or grouped hierarchically?
   Can resources be tagged on different hierarchy levels?
   How are connections created?
   Automatic Tagging Is tagging enriched with automatically created tags and relations (for instance file types, automatic expansion of terms etc.)?