EEXCESS
Enhancing Europe’s eXchange in Cultural Educational and Scientific reSources

Deliverable D4.4

Integration and Enrichment Services Final Prototype

Identifier: EEXCESS-D4.4-JR-DIG-Integration-and-Enrichment-Services-Final-Prototype_V01.doc
Deliverable number: D4.4
Author(s) and company: Thomas Orgel, Werner Bailer, Martin Höffernig, Werner Preininger, Silvia Russegger (JR-DIG)
Internal reviewers: Kris Jack (MEN)
Work package / task: WP4
Document status: final
Confidentiality: Public
Version: 2010-07-12
History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Reason of change</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2016-01-26</td>
<td>Document created (e.g. structure proposed, initial input...)</td>
</tr>
<tr>
<td>2</td>
<td>2016-02-26</td>
<td>Input on metadata quality</td>
</tr>
<tr>
<td>3</td>
<td>2016-03-18</td>
<td>Update PartnerWizard, Metadata Quality Assessment</td>
</tr>
<tr>
<td>4</td>
<td>2016-04-29</td>
<td>Metadata Quality Assessment</td>
</tr>
<tr>
<td>5</td>
<td>2016-06-29</td>
<td>Version for internal review</td>
</tr>
<tr>
<td>6</td>
<td>2016-07-12</td>
<td>Final Version</td>
</tr>
</tbody>
</table>

Impressum

Full project title: Enhancing Europe’s eXchange in Cultural Educational and Scientific reSources
Grant Agreement No: 600601
Workpackage Leader: Thomas Orgel, JR-DIG
Project Co-ordinator: Silvia Russegger, JR-DIG
Scientific Project Leader: Michael Granitzer, Uni-Passau

Acknowledgement: The research leading to these results has received funding from the European Union’s Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 600601.

Disclaimer: This document does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of its content.

This document contains material, which is the copyright of certain EEXCESS consortium parties, and may not be reproduced or copied without permission. All EEXCESS consortium parties have agreed to full publication of this document. The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the EEXCESS consortium as a whole, nor a certain party of the EEXCESS consortium warrant that the information contained in this document is capable of use, nor that use of the information is free from risk, and does not accept any liability for loss or damage suffered by any person using this information.
# Table of Contents

1 Executive Summary .......................................................................................................................... 1

2 Introduction .................................................................................................................................... 2  
   2.1 Purpose of this Document ........................................................................................................ 2  
   2.2 Scope of this Document ............................................................................................................ 2  
   2.3 Status of this Document ........................................................................................................... 2  
   2.4 Related Documents ................................................................................................................ 2

3 Problem to solve – User scenario .................................................................................................... 3

4 Solution Components ....................................................................................................................... 4

5 PartnerWizard .................................................................................................................................. 9

6 Refinement Mapping Configuration .................................................................................................. 18  
   6.1 Definition of mappings based on compound concepts .............................................................. 18  
   6.2 Support of round trip mappings ............................................................................................. 20

7 Metadata Quality Assessment ........................................................................................................... 21  
   7.1 State of the Art ........................................................................................................................ 21  
   7.2 Assessing Source Data Quality ................................................................................................ 22  
      7.2.1 Quality measures .............................................................................................................. 22  
      7.2.2 Representation of data quality measurements ................................................................. 24  
      7.2.3 Implementation ............................................................................................................... 27  
      7.2.4 Results ............................................................................................................................ 29  
   7.3 Assessing Transformation and Enrichment Data Quality ........................................................... 36  
      7.3.1 Metrics and Implementation ......................................................................................... 36  
      7.3.2 Results ............................................................................................................................ 36  
   7.4 Refined Mapping Quality Assessment ...................................................................................... 38

8 Conclusions ...................................................................................................................................... 42

9 References ....................................................................................................................................... 43

10 Glossary .......................................................................................................................................... 44
1 Executive Summary

EEXCESS aims to bring cultural, scientific and educational content to the user. This means that the user gets additional information within the environment the user is using and working with. EEXCESS aims to unfold the treasure of cultural, scientific and educational content to improve the interconnectedness between these different domains.

One important achievement of EEXCESS is to provide not only software components but also working prototypes. Besides a Chrome browser plugin that enables to list additional cultural and scientific information while looking at and editing a Wikipedia article some more software add-ons and plugins were developed – e.g. WordPress plugin for editing WordPress blogs, GoogleDocs Plugin to include resources and citations in the right way (all described in D2.5 Final Software Components for Presentation and Augmentation Interfaces).

To get the results of different institutions listed the following tasks have to be fulfilled by content provider:

1) Create a mapping of local data to the EEXCESS data format
2) Include the data in the EEXCESS Framework by providing a PartnerRecommender on the local system that is integrated in the Federated Recommender of EEXCESS

To provide tools for both these tasks work package 4 of EEXCESS is foreseen. Both have already been documented and described in D4.3 so refinements are described within this deliverable.

First versions of features to fulfil these tasks were provided in autumn 2014 during the first testbed and evaluation phase. Especially the testbed phase leads to new awareness regarding the usability and therefore also new features and a new tool was planned and implemented.

One feedback from the evaluation phase was that the establishment of a PartnerRecommender seems to be rather complicated as it was based on programming knowledge. After conceptualisation and planning within the last project year we implemented a web based application the so called PartnerWizard. This tool enables the user to create her own PartnerRecommender without deep going implementation skills. By “simply” enter configuration parameters into a web formula a first PartnerRecommender is built that can be tested against results of the specific partner’s data. The user herself can decide if changes should be done in order to make the result list better. Once the partner is satisfied the PartnerRecommender can be deployed on the development server of the EEXCESS consortium. The shift from the development server to the stable – and so public – server is done by the EEXCESS consortium in order to have control which data are available via EEXCESS and to make sure that the data access agreement is signed by the new partner.

Having more and more cultural information on the web makes it rather difficult for users to assess the quality of the data she receives. On the other hand professional data providers need to and want to get information regarding their data quality. And last but not least it is very important for systems like EEXCESS or portals to provide high quality data – even after enriching the results by background information.

To meet these requirements we investigated in answering questions like “How can the heterogeneous information from various providers be harmonised to give the user a uniform data access?” , “How can the result list be made better?” or “What has to be done to get my results listed and ranked as well?” EEXCESS provides different tools and features to give answers to these questions:

1) Assessment of Mapping Quality: With this tool the data provider gets feedback regarding the mapping per se. Is there information that is going to be lost via the transformation process?
2) Assessment of Metadata Quality: To get feedback regarding the metadata quality of a data provider we developed an application publically available as web application and desktop application to make quality assessment on XML meta data files. After uploading the XML metadata file the application starts a detailed quality assessment – described within this deliverable. The results are listed in an output html-file that can be analysed by the data provider later on.

Starting with a concrete user scenario we describe the process chain for a data provider to become partner of the EEXCESS data provider community and to how the data provider benefits from the EEXCESS tools and application in making his data wider available and in improving the data quality in his local system. In this regard we include a section describing the state of the art of metadata quality assessment where we especially took Europeana – as the biggest hub for cultural data – into account to show the actual status and the next planned steps.
2 Introduction

2.1 Purpose of this Document
This deliverable provides the final prototypes for the integration of cultural, scientific and educational data sources as well as the enrichment of those data sources from social media channels and for performing quality assessment regarding the data mapping as well as the source meta data in local information systems.

2.2 Scope of this Document
Deliverable 4.4 will give an overview on the different components of the EEXCESS Framework for data provision. It will also list and describe the necessary tasks that have to be fulfilled by different content providers to get data included into the EEXCESS framework.

The document describes the improvements in development and implementation which was based on the results of the first and second deployment and testbed phase of EEXCESS. Based on a defined use case (see chap. 3) the description in D4.4 should enable a new data provider to include her digitised objects, documents and images into the EEXCESS framework as well as to perform quality assessment on the mapping of data and on the source meta data per se.

2.3 Status of this Document
This is the final version of D4.4. The project internal review was done by MEN.

2.4 Related Documents
Before reading this document it is recommended to be familiar with the following documents:
- D1.2 Second Conceptual Architecture and Requirements Definition
- D1.3 Third Conceptual Architecture and Requirements Definition
- D4.1 Integration and Enrichment Specifications and Analysis
- D4.2 First Integration and Enrichment Services Prototype
- D4.3 Second Integration and Enrichment Services Prototype
- D3.3 Second Federated Recommender Prototype
- D2.5 Final Software Components for Presentation and Augmentation Interfaces
3 Problem to solve – User scenario

**Persona:** Helmut J.

**Education:** programmer

**CV:** After high school Helmut started working as an IT Administrator for a local museum.

**Scenario Description:**
The director of the Museum where Helmut works has heard of EEXCESS and wants to bring in the content of his museum into the EEXCESS ecosystem. So he gives Helmut the task to offer the content to EEXCESS. Helmut browses the EEXCESS website and finds different ways to act as a data provider for EEXCESS. The Museum offers a searchable web site with the content of the museum, which also provides access via an API. So he decides to use the PartnerWizard provided by EEXCESS.

By generating the transformation file Helmut realises that it would be useful to learn details regarding the data quality and the structure of his data. By browsing the EEXCESS homepage Helmut also finds out that the consortium has invested huge effort in features to gather feedback regarding the data structure and quality of the data. As Helmut wants to use the EEXCESS PartnerWizard he will also use the quality measure features of EEXCESS.

**Role of EEXCESS:**
EEXCESS provides a template which the user can use to create a new PartnerRecommender. EEXCESS provides all necessary libraries in a public repository.

EEXCESS offers a WebApp the so called PartnerWizard to build a new PartnerRecommender using the template to bring in content that is available via an API.

EEXCESS also offers services where the new data provider can hook into the EEXCESS ecosystem and services to optimise the results (work fulfilled in work package 3).

EEXCESS offers features to gather feedback regarding the data quality (metadata quality and mapping quality) as quality assessment.
4 Solution Components

To understand how we can integrate the new data provider into the EEXCESS Framework we give a brief overview over the EEXCESS architecture. To explain how the EEXCESS Framework works, we start at a client where a user wants to retrieve information in. For our first use case in EEXCESS the “client” is a Chrome browser, but this special client can be replaced by any other type of client (e.g. learn management system, GoogleDocs, WordPress). The following list gives an overview on the different tasks right from the start of a user driven search process to the final result presentation.

- The client detects an information need and creates a query.
- The client sends this query to the Privacy Proxy. The Privacy Proxy invokes the Federated Recommender. This component has the information which data provider are actually available.
- The Federated Recommender calls all the PartnerRecommender
- PartnerRecommender sends results back.
- The results are integrated in one combined view and are sent back to the client via the Privacy Proxy.

A more detailed description of the components and their interaction is provided in D1.2 & D1.3.

To add a new data provider we need to build a new PartnerRecommender which queries the partner data store and returns the results in the EEXCESS data format to the Federated Recommender.

The PartnerRecommender needs to run on a Apache Tomcat\(^1\) or another Java Servlet Container.

The following figure gives an overview on the EEXCESS components as specified and described in detail in D1.1 First Conceptual Architecture and Requirements Definition.

\(^1\) http://tomcat.apache.org/
Not directly part of the EEXCESS architecture but important application part especially for data providers is the so called “PartnerWizard”. The PartnerWizard is a stand-alone Web application that offers the possibility to create a PartnerRecommender without being a software development specialist. Input and output specifications that are necessary to have a valid PartnerRecommender are defined via available configuration forms.

From the view of architecture diagrams the application stands outside of the EEXCESS architecture that returns a ready to use partner recommender. The following chapter explains the usage and necessary configuration in detail.
A second application outside the core EEXCESS architecture is the “ConfigTool”. It was already explained in D4.2 and was further developed as specified and foreseen in D4.2. Via the ConfigTool the necessary metadata mapping from the source data format from the data provider repository to the EEXCESS data format can be defined. Output is a transformation file that now is directly used within the EEXCESS PartnerRecommender to be able to display and use the “local” data within the EEXCESS features. Refinement decided upon during the first testbed and deployment phase are described further on.

Figure 2: Overview components

Figure 2 shows how the data quality tool gets the data for checking the quality. We have developed a simple framework, shown here as Eval Framework. This framework gathers log files and keywords from EEXCESS Server, used in the EEXCESS Framework and combines these keywords to a list without duplicates. Further on the framework creates smaller, randomized set of keywords. These keywords-sets are going to be used as input to query our PartnerRecommender in a local setting. These PartnerRecommender queries APIs of the data provider and writes the results, the transformed results and also the enriched results in LogFiles. Later on the EEXCESS data quality tool uses these LogFiles as an input, analyses them, and generate the Data Quality Report. As described in D1.3 Figure 3 shows the workflow of the recommendation method. The PartnerRecommender is invoked by the Federated Recommender with the User Profile. The PartnerRecommender generates a query for the partner data store. This call returns the results in the format of the partner system. The PartnerRecommender takes this result and gives this to the Transformer. Using a mapping transformation file (XSLT File and produced with the EEXCESS ConfigTool) the Transformer produces a result file in EEXCESS data format. The PartnerRecommender returns these results to the Federated Recommender.
In Figure 3 the workflow recommendation call is shown. The PartnerRecommender gets invoked by the Federated Recommender with a list of objects. The PartnerRecommender generates calls to gather details to this objects for the partner data store. These calls return results in the format of the partner system. The PartnerRecommender takes these results and gives them to the Transformer. The Transformer generates the EEXCESS-format as described above. After the transformation the results are in the EEXCESS-format. The PartnerRecommender takes these results and calls the enrichment component and then returns the enriched results to the Federated Recommender.

In Figure 4 the workflow of the details call is shown. The PartnerRecommender gets invoked by the Federated Recommender with a list of objects. The PartnerRecommender generates calls to gather details to this objects for the partner data store. These calls return results in the format of the partner system. The PartnerRecommender takes these results and gives them to the Transformer. The Transformer generates the EEXCESS-format as described above. After the transformation the results are in the EEXCESS-format. The PartnerRecommender takes these results and calls the enrichment component and then returns the enriched results to the Federated Recommender.

As pointed out in D4.3 we changed the process of the enrichment to get acceptable response times. In order to get acceptable response times, the PartnerRecommender works with a thread pool to process the list of objects in the detail call. In addition we have added a timeout mechanism to prevent a single processing of one record to block the whole system. So if the timeout occurs, the PartnerRecommender skips those records.
which took too long and returns the others. The actual running tests of the deployed PartnerRecommender show, that this approach works well and with acceptable response times.

We have made several changes in the base components which are used by the hand made PartnerRecommenders and also used by the PartnerRecommenders build with the PartnerWizard. Also the implementation of supporting JSON response within the PartnerWizard leads to some changes and adoptions in the base components of the PartnerRecommender. So we extended the interface of the PartnerConnectorApi with new methods for setting the format of the API responses to JSON or XML. These methods are used in the base PartnerRecommender implementation. If the format is JSON we transform the JSON to XML in the base implementation and then work with the transformed XML.
5 PartnerWizard

In the following section we want to point out how Helmut J. - our IT Administrator from the use case - can easily create a PartnerRecommender and so provide his data to the EEXCESS framework and his users. In detail the necessary steps are already described in D4.3. We have updated the description here including the changes we made also in respect to new features.

For the success of EEXCESS the number of data provider is a very important measurement. Therefore, it seems critical to have good and easy to fulfil procedures or necessities to get especially data provider from the cultural domain on board. We decided to publish a guide how new APIs from new data provider can be included. This was already done during the testbed and evaluation phase in year 2 and 3.

One result of the evaluation of this guide was that the data provider needs a person with implementation skills to get the PartnerRecommender running. Many of the data providers cannot implement this method on their own. In order to become more attractive for long tail content providers, which EXCESS wants to address, we decided to offer a second easier to handle method – the so called PartnerWizard.

With the PartnerWizard we have improved the process of integration by reducing the steps and complexity of this process. The build system we use to build the system for the Server side components of the EEXCESS Framework is Apache Maven\(^2\). With Maven it is possible to build the components only using the source and the Maven configuration; no additional action is required, like downloading libraries, etc. Maven provides a feature called archetype, which enables the developer to provide templates for a software project. We have created such a Maven archetype which enables the developers to create the structure for the new PartnerRecommender with one single command. This command needs specific parameters to create the PartnerRecommender.

\(^2\)https://maven.apache.org/
Figure 5: Maven archetype partner-recommender

Figure 5 shows the structure of the Maven archetype. The sub-folder “src/main/resources/archetype-resources” contains the root of the template. The other files and folders are needed by the maven archetype. In this structure the parameters which the archetype can handle are defined.

During the generating process maven replaces placeholder in file names (surrounded with __) and in files with the values passed from the call of the archetype. Figure 6 shows an example source file with the placeholders and how they are used in the archetype.

```java
cpyackage ${package}.recommender;

import eu.eeexcess.partnerrecommender.api.PartnerConnectorApi;

/**
 * Query generator for ${partnerName}.
 */
public class ${artifactId}PartnerConnector extends PartnerConnectorBase implements PartnerConnectorApi {
}
```

Figure 6: maven archetype partner-recommender - source code
All necessary libraries are included and are accessed from the public repository.

The source of the archetype is published on Github\(^3\). The compiled archetype is published on the public repository of the EEXCESS partner KNOW Center\(^4\).

Further on the next step of improvement was that we created a web-application hosted on our servers where the data provider can use simple web page forms to fill out some specific parameters to create the new PartnerRecommender.

Figure 7 shows a web form to enter some general information needed to build a valid Java project and also information needed for registration into the EEXCESS framework.

**PartnerWizard**

The PartnerWizard allows you to integrate new partner systems using this web app. A partner system is a content provider which is used by the federated recommender to generate recommendations. So without coding you can integrate most search APIs. For more general information about the EEXCESS project please visit [http://eexcess.eu](http://eexcess.eu) or [https://github.com/EEXCESS/PartnerWizard](https://github.com/EEXCESS/PartnerWizard)

For more information about this web app please visit: [https://github.com/EEXCESS/PartnerWizard](https://github.com/EEXCESS/PartnerWizard)

![Figure 7: PartnerWizard - general parameters](image)

Figure 7 shows the form to define the URL of the API for the search call. There is also an additional field for an example search term. With the button “Call API Search” the WebApp, generates the service call and executes it. The response will be shown below in the section API response. When the user gets a response he must define the XPath where the single results are located in the response and then define mappings to the defined EEXCESS fields. These mappings must be entered as XPaths, relative to the upper defined XPath for the loop. There is also a button “test” where the user can probe the entered XPaths on the actual result. The results of this probe will be shown in the right column in this table.

\(^3\) [https://github.com/EEXCESS/PartnerWizard/tree/master/archetype-partner-recommender](https://github.com/EEXCESS/PartnerWizard/tree/master/archetype-partner-recommender)

\(^4\) [https://nexus.know-center.tugraz.at/content/repositories/eexcess/](https://nexus.know-center.tugraz.at/content/repositories/eexcess/)
configure Search API

The API configured in the sections below need to return the data in XML-format. EEXCESS uses XPath to determine the metadata from the objects. If you need more info about XML and XPath we recommend to visit:

- http://www.w3schools.com/xsl/path_intro.asp
- http://www.w3schools.com/xml/xpath.asp
- http://mitbox.pop.gu.se/path_generator.html
- http://www.xmltest.com/xpath

Figure 8: PartnerWizard - parameters for search API

The entered search term will be used to generate automatic JUnit-Tests in the source project which the WebApp generates.
A similar web form is also available to configure the detail call API. Figure 9 shows the parameters for configuring a detail call to the API to get more detailed data for the objects, to make best use especially in the visualization tools of EEXCESS and in the content creation components (WordPress plugin, Moodle plugin). As search term a unique identifier is required in the field search term. This value must be the corresponding value as defined in the mapping from the search result for the field ID.

Figure 9: PartnerWizard – parameters for detail API

Once the user has finalised the configuration and is satisfied with the result the new PartnerRecommender can be created with the button “Build the PartnerRecommender”. This is shown in Figure 10. If it works well, the
WebApp shows the Maven logs. The WebApp computes the entered parameters to a Maven command show in Figure 10. The user can also execute the command on his own machine.

Create PartnerRecommender/project

mvn command used to generate the PartnerRecommender:


Result:

20160616_155717
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.

EEXCESS.github.com:C:~/avalon-tomcat-7.0.43:EEXCESS/bin-cd:C:dev/eexcess-partnerrecommender-archetype-sandbox

EEXCESS.github.com:C:dev/eexcess-partnerrecommender-archetype-sandbox:SQRKerlingRecommender

EEXCESS.github.com:C:dev/eexcess-partnerrecommender-archetype-sandbox:SOLRKerlingRecommender:mvn clean install -DskipTests

[WARNING] Some problems were encountered while building the effective model for at joanneum.eexcess-partner-SQRKerlingRecommender war:1.0-SNAPSHOT
[WARNING] 'dependencies.dependency'(group artifact type classifier) must be unique. eu.excess.eexcess-reference-partner-data-layer jar -> version 1.0-SNAPSHOT vs 1.0-SNAPSHOT -Drelease version) @ line 70, column 21
[WARNING] 'build plugins plugin version for org.apache.maven.plugins.maven-compiler-plugin is missing. @ line 107, column 12
[WARNING] 'build plugins plugin version for org.apache.maven.plugins.maven-complex-plugin is missing. @ line 111, column 12
[WARNING] It is highly recommended to fix these problems because they threaten the stability of your build.
[WARNING] For this reason, future Maven versions might no longer support building such malformed projects.

Download PartnerRecommender as WAR
Download the generated sources of the PartnerRecommender

Figure 10: PartnerWizard - build PartnerRecommender

On the bottom of the screenshot there are two links where the user can download the compiled WebApp and deploy it on any Tomcat server which has access to the the API and to the EEXCESS-Framework. With the other link the user can create a zipped package of the generated source files and download this. This enables the user to use this source code as a base for further changes or improvements.

The PartnerWizard can deploy the new generated PartnerRecommender to our development environment and it will automatically register at the FederatedRecommender on our development server. So, if the user has downloaded the EEXCESS-Chrome-Extension for the development environment, the user should see recommendations from his data store. The URL for this version of the EEXCESS-Chrome-Extension is mentioned in the WebApp.
Figure 11: PartnerWizard – Deployment – Query Generation Configuration

- src/test/java
  - (default packages)
    - SOLRkierlingRecommenderPartnerRecommenderTest.java
- src/main/java
  - at.joanneum.datalayer
    - SOLRkierlingRecommenderTransformer.java
  - at.joanneum.recommender
    - SOLRkierlingRecommenderPartnerConnector.java
    - SOLRkierlingRecommenderQueryGenerator.java
  - at.joanneum.webservice.tool
    - PartnerStandaloneServer.java
- src/main/resources
  - log4j.properties
  - mapperObject.xsl
  - mapperResultList.xsl
  - partner-config-query-template.json
  - partner-config.json

Figure 12: generated sources

Figure 12 shows the structure of the generated sources of the PartnerRecommender.

Figure 13: PartnerWizard - configuration

Figure 13 shows the generated configuration of the PartnerRecommender. The parameter FederatedRecommenderURI points to the endpoint of the FederatedRecommender where the new PartnerRecommender registers. In this case it points on the same machine, because the PartnerWizard is deployed on the development server.
The new PartnerRecommender is already configured to connect to the FederatedRecommender on the same server, so that after the PartnerRecommender is deployed via the PartnerWizard and has registered on the FederatedRecommender, the new data source will be used.

For testing the PartnerWizard we have created two new PartnerRecommenders. One for the RijksMuseum and another for a local museum named Museum Kierling. We have also tested the PartnerWizard with an already existing Partner, KIMPortal.

The KC has developed a WebApp, which enables the user to optimise the results by probing different strategies for generation of the queries to the PartnerAPI. We already have combined these two applications into one WebApp. After the PartnerWizard has created the first working version, the system part from JR-DIG provides the generated configuration to the part developed by KC and the WebApp switches to the second page, where different strategies for the Query Generation will be tested. If this step is finished the resulting configuration will be passed back to the PartnerWizard and the new PartnerRecommender will be generated with the new configuration.

Figure 14 shows how the PartnerWizard, and how his components are related to each other.

We have build several PartnerRecommenders with the actual prototype and they are already deployed on our Development Server. Concrete we have added the following data providers to our Development Environment:

- Digital Public Library of America
- DeutscheNationalbibliothek
- The National Archives UK
- RijksMuseum
- Museum Kierling
- CORE.ac.uk
- Swissbib
- Landesarchiv Steiermark
• Landesarchiv Kärnten

The PartnerWizard generates Log-Files, which enable us to rebuild the generated PartnerRecommender with a framework for software testing of web applications. So the PartnerWizard generates script files which we can use with the Selenium Framework\(^5\).

6 Refinement Mapping Configuration

This section describes new extensions of the metadata mapping configuration tool in order to improve the functionality and user experience. These extensions concern support for defining templates involving compound concepts and editing the context information and structure of the input format in order to support round trip mappings needed for the assessment of the mapping quality.

6.1 Definition of mappings based on compound concepts

The definition and editing possibilities of mappings based on compound concepts have been improved. A compound concept describes a group of metadata elements containing at least two metadata elements. For example, a place (compound concept) can be defined by its latitude and longitude (both atomic concepts). The definition of classes and properties required for describing these relations are presented in Section 4.2 of D4.1. These are the classes `meon:CompoundConcept`, `meon:AtomicConcept` and the property `meon:contains`.

However, defining such a relation between a compound concept and contained concepts is only done on a conceptual level. Managing the configuration of concrete mapping templates based on compound concepts in the metadata mapping configuration tool was not possible since two required configuration features were not available. In the first place, defining the data type representation (XML structure, data type and XPath information) of a contained concept in context of the data type representation of its compound concept was missing. Secondly, a solution for specifying the position of the mapping result of a contained concept in the corresponding output structure of its compound concept was also not available.

These missing configuration features are implemented in the current version of the metadata mapping configuration tool. Referring from the data type representation of a concept to the data type representation of its compound concept is done by setting the context to the compound concept. For example, in Figure 15 the concept Latitude is part of the compound concept PlaceTimeEvent. This relation is also reflected in the data type representation of concept Latitude by setting the value of field context to PlaceTimeEvent.

![Figure 15: Data type representation of concept Latitude in context of compound concept PlaceTimeEvent.](image)

In addition, a new data type named MixedContent applicable for compound concepts is introduced. Using this data type, the position of the mapping result of a contained concept in the output structure of its compound concept is specified. Therefore a special tag name `<call-template.NameOfConcept>` as a placeholder is used. For example, in Figure 16, the definition of a data type representation of the compound
concept PlaceTimeEvent based on the MixedContent data type is depicted. This compound concept contains three concepts, namely Latitude, Longitude, and Time. The position of the mapping results of these concepts are defined by the tags `<call.template.Latitude>`, `<call.template.Longitude>`, and `<call.template.Time>` in the output structure.

![Figure 16: Data type representation of compound concept PlaceTimeEvent using MixedContent data type.](image)

In Figure 17, an example of a mapping result of the compound concept PlaceTimeEvent is depicted. The mappings for the concepts Latitude, Longitude, and Time are performed and replace the placeholder tags in the output structure. For example, the concept Latitude is mapped based on the data type representation presented in Figure 15. Thus the tag `<wgs84_pos:lat>` including the value of the mapping result as a string replaces the placeholder tag `<call-template.Latitude>`.
6.2 Support of round trip mappings

In order to perform the assessment of the mapping quality (cf. Section 7.4), round trip mappings have to be defined. During a round trip mapping a metadata document is first mapped to a given target format and then mapped back to the original metadata format. Therefore mappings from the output format back to the input format also have to be defined.

Expressing equivalent mapping relations on the conceptual level is possible since the initial version of the metadata mapping configuration tool. In addition, it is also required to specify XPath locators in the output format and enable the definition of the XML structure for the input format. These features are implemented in the current version of our tool.
7 Metadata Quality Assessment

This section provides a brief update on recent related work related to metadata quality assessment. Then the results on assessment of source metadata quality (i.e., data as made available by the data providers), metadata quality after enrichment and for assessing mapping quality are presented.

7.1 State of the Art

A review of the state of the art has been part of D4.2 and D4.3, and the related work discussed there is not repeated in this document. However, assessment of metadata quality has become a quite active topic in different projects and initiatives dealing with cultural heritage and scientific content (e.g., Europeana defines data quality as one their priorities for the coming five years\(^6\)). Thus a summary of recent related work is presented in the following.

In the context of Europeana and the Public Digital Library of America (DPLA\(^7\)) work on analysing the quality of contributed and enriched data records has been performed. The majority of the work deals with measuring completeness of records and provides statistics on related metrics, such as coverage of mandatory elements.

This starts with basic statistics\(^8\) such as those about objects in the recently established Europeana Publishing Framework\(^9\). As the tiers defined by the publishing framework imply different degrees of completeness of the records, these statistics provide some insight about the completeness of the contributed data. Europeana currently has a completeness measure in place\(^10\), which yields a rating on a 11 point scale. Statistics about the number of items in each of the completeness have also been collected\(^11\). A quite comprehensive analysis of data on Europeana with a focus on completeness has been performed by Király\(^12\), and the source code of the tool implemented to perform the analysis is available.

A tool called MoRe Quality to detect certain types of quality issues has been implemented based on the framework of the LoCloud\(^13\) project, and is available as open source software\(^14\). The tool detects errors such as invalid ISO 8601 date formats, invalid/unresolved hyperlinks, invalid language codes and not well-formed author names. The tool allows the extension of the tool to other measures.

In the context of DPLA, the completeness of metadata records and their impact on usage patterns has been researched\(^15\). It includes statistical analysis of presence of certain field types, but also analysis of some filed value properties such as use of subjects and use of extended date/time formats. In addition, outliers have been identified by these analyses, e.g. artefacts of ingesting collection-level data together with record-level data. One interesting finding is that record view counts are not correlated with the number of data provided, which is explained that most viewers view items at the providers’ pages rather than on DPLA.

In early 2015, Europeana has established a Data Quality Committee in order to analyse quality issues and work on appropriate quality metrics. The group has started working on a document for discovery and metadata scenarios. Issues that will be initially dealt with include mandatory fields in EDM, completeness measures, and issues with field values. EEXCESS has contributed to these efforts with results from quality analysis of data used in the testbeds originating from Europeana.

---

\(^6\) http://pro.europeana.eu/page/data-quality-etch15-roundtables

\(^7\) http://dp.la/

\(^8\) https://docs.google.com/spreadsheets/d/10hdB3nRhcjblCICgFNSnTN12FmJylup_HYRIffH_3tbHM/edit#gid=1682075788

\(^9\) http://pro.europeana.eu/publication/publishing-framework

\(^10\) https://docs.google.com/document/d/1Henbc0lQ3gerNoWudJ5DCpNq4YkOxOW55Q7g4f2bP0/

\(^11\) https://docs.google.com/spreadsheets/d/l1_MVILp8JG1s1UEiQidGkijVoR6u1r1ucz-qwwo/edit?verb=2&pli=1#gid=1273225606

\(^12\) http://pkiraly.github.io/2016/01/15/second-report/

\(^13\) http://www.locloud.eu

\(^14\) https://bitbucket.org/vbanos/more-quality/

\(^15\) https://dplafest2015.sched.org/event/2UfQ/can-metadata-be-quantified-analytics-for-libraries
7.2 Assessing Source Data Quality

In contrast to other related work (e.g. [Bellini, 2013, Gavrilis, 2015]) we cannot assume a single application profile (i.e., a single definition from one organisation of how a metadata format is used) in the EEXCESS system, against which completeness and accuracy can be checked, but the profile to be applied will depend on the type of content and the data provider (using the knowledge about the respective native data model).

In this section, we describe the implemented quality measures, we describe the representation of the quality metrics and we present some results of applying them to data gathered during the testbed.

Finally, this section discusses the representation and visualisation of the obtained quality metadata.

7.2.1 Quality measures

We discuss the different metrics grouped by dimensions of metadata quality, following the dimensions defined in [DQV, 2015]. For some of the dimensions, no metrics have been implemented. Accuracy is difficult to measure automatically, especially, as a digital representation of the object being described may not be accessible. Conformance can be checked by validation against a schema, where available, thus no specific metric has been implemented. Relevance cannot be measured beyond completeness. Timeliness cannot be automatically assessed in this context.

Statistics

This set of measures concerns basic quality metrics as described in [Bellini, 2013], such as counting the number of records provided, the number of empty records, and normalising these numbers w.r.t. the number of records in the data from different providers.

Availability

We have in particular addressed the accessibility of controlled vocabularies referenced from the data records. We analyse whether references to controlled vocabularies using URIs are used in the data. For interoperability and linking with other data is important that the terms of the vocabulary are accessible, i.e., are identified with URLs that can be resolved. Ideally, this URI does not only resolve to a human readable description, but to a machine readable definition, which can be used to relate the term to other data sources. We use content negotiation to perform this analysis. The basic idea of content negotiation is to serve the best variant for a resource, and to serve it based on:

- What variants are available, and what variants the server may prefer to serve
- What the client can accept, and with which preferences: in HTTP, this is done by the client which may send, in its request, Accept headers (Accept, Accept-Language and Accept-Encoding), to communicate its capabilities and preferences in Format, Language and Encoding, respectively 16.

Agent-driven negotiation is realised by analysing the response of the server after receiving an initial request to the resource. We analyse the possibilities of content negotiation regarding the URIs in the dataset. The idea is to use agent-driven negotiation to gather information about the variants a server can serve a resource behind an URI. Possible variants may be e.g. RDF/OWL, XML, JSON or plain text.

Completeness

The completeness metric in EEXCESS is based on the statistics of returned fields and fields containing values:

$$\text{completeness} = \frac{\text{fields returned}}{\text{fields total}} \cdot \frac{\text{fields populated}}{\text{fields returned}}$$

As the EEXCESS model does not explicitly define mandatory fields, none of the fields is treated specially in this metric. One issue is determining the number of total fields in models where multiple occurrences of fields are permitted. In the simplest case, each type of field is counted only once. However, if it is considered beneficial to have multiple occurrences of a field (e.g. multiple subject classifications), a higher value than one can be used to favour records providing more than one instance.

16 https://www.w3.org/blog/2006/02/content-negotiation
Processability
Processability measures the degree to which information is appropriately structured and well-defined to be automatically processed. We considered the structuredness of values (e.g., conforming to patterns) and the machine readable structure of rights metadata.

Structuredness
We determine measure about the structuredness of values, for example of fields containing dates, names or dimensions of objects. The aim is not only to make a binary decision whether they are structured, but also whether the format the field can be inferred (e.g., using regular expressions).

In order to determine if a field is structured we extract a regular expression from all values of the respective field in a data set.

A large part of the related literature relies on labeled training set to learn regular expressions. This is not practical for our application, thus we only consider approaches that allow unsupervised extraction of regular expression. The algorithm needs to work only on positive data, however, the positive samples may be noisy. Such an approach with the application to inferring DTDs from XML documents has been proposed in [Bex, 2010]. [Fernau, 2005] proposes an algorithm which serves our purpose well, starting from grouping characters by type and then inferring a tree structure of a regular expression. [Li, 2008] propose an approach that is guided by a prototype regular expression. This can be useful for some types of fields, where assumptions about possible patterns can be made. [Bartoli, 2012] propose an approach based on genetic programming, however, it may be computationally too expensive for our application.

Our approach contains several steps, which aim to speed up the process by performing extraction of regular expressions only if there is a high likelihood for actually having a common structure in the field values.

We preprocess field values by pruning white spaces. As a first measure, we determine a histogram of field lengths over the data set. Data with well-defined structure (e.g., dates, lengths) will show clear peaks in the histogram, which is an indicator for the structuredness of the field. We then detect characters such as hyphens, commata and periods which are also indicators for structured data. We also scan the fields for the presence of SI unit abbreviations

We then implement the first step of the approach described in [Fernau, 2005], i.e., block-wise grouping and alignment. In this step, each character or digit gets replaced by an indicator of its type. We can then count the number of times each pattern appears in the data. Let \( N \) be the number of patterns found, and \( p_i, i=1,...,N \) be the number of each of the patterns occurs. We then determine a structuredness score as the complement of the fraction of distinct values and samples, i.e.

\[
\text{structuredness} = 1 - \frac{N}{\sum_{i=1}^{N} p_i}.
\]

Higher values correlate with lower variation in terms of patterns. This score yields a similar trend than the median count of patterns or the sum of the k most frequent patterns, but is more sensitive to repeated values and thus provides better discrimination between structured and unstructured fields.

Machine readable rights metadata
The approach for vocabulary quality assessment described above can also be applied to address some quality aspects of rights metadata, in particular, to determine if rights statements contain only free text or references to machine readable licenses. Similar to vocabularies, it can be checked if the license statement is accessible. In addition, it can be checked whether the license statement is one from a set of known statements, such as from the Creative Commons family of licenses.

Credibility
Credibility could be measured using provenance metadata, but while the EEXCESS data model supports provenance information, it is not available from the data providers, so that it cannot be used to assess credibility of source metadata. We thus limit credibility measurements to analyzing outliers.
We use the frequencies of patterns determined by our structuredness score. We assume that the distribution of patterns is governed by a normal distribution $\mathcal{N}(\mu, \sigma)$. We determine frequent and rare outliers as

$$\text{outlier}_r = \frac{1}{N} \sum_{i=1}^{N} \left\{ \begin{array}{ll} 1, & \text{if } f_i > \mu + 2\sigma \\ 0, & \text{otherwise.} \end{array} \right.$$ 

$$\text{outlier}_f = \frac{1}{N} \sum_{i=1}^{N} \left\{ \begin{array}{ll} 1, & \text{if } f_i < \mu - 2\sigma \\ 0, & \text{otherwise.} \end{array} \right.$$ 

For fields with a high structuredness score we are interested in the rare types of patterns. For example, in a field with language codes most values will be a two or three character string, while less frequent patterns are likely to be indicators of problems. For fields with a low structuredness score, we are interested in the frequent types of patterns. For example, in a title field, most patterns will occur once or a few times, while a more frequent pattern may indicate a systematic error or default value being used. Thus we determine an outlier score weighted by structuredness:

$$\text{outlier} = \text{outlier}_r (1 - \text{structuredness}) + \text{outlier}_f \text{ structuredness}$$

For fields using controlled vocabularies, we can apply the same metrics as for patterns also to the actual values. This gives an indication of the homogeneity of values used, and may indicate misspelled or correct but rarely used values.

**Consistency**

We use a consistency metric to determine the diversity of terms in specific fields (e.g., subjects) of a record based on similarity metrics calculated on WordNet. The idea of the metric is to determine how related the terms found in the annotation are. Such a metric can never be absolute, as the range of valid annotation varies considerably between different types of objects (e.g., a single spoon vs. an encyclopedia) and on the annotation practices applied. Thus an important use of the consistency metric is it comparative application before and after enrichment, i.e., to determine whether the enrichment has significantly reduced the accumulated similarities of the terms in the annotation. Enrichment usually complements available annotation by adding broader, narrower or related terms, but hardly introduces a completely new aspect. Thus a strong decrease in the term similarities is likely an indicator of an error introduced during enrichment.

The consistency metric for a record can be calculated from a single or multiple fields of the record. If applicable (e.g., for titles), stopwords and short words are removed. Of the remaining words, only nouns are kept. This results in a set of nouns $\mathcal{N}=\{n_1, \ldots, n_k\}$ for the record. We use a similarity metric $d(n_i, n_j)$ that provides a score for the distance of two nouns in WordNet, based on the lengths and types of relations between them. A number of such metrics exist, and in our implementation we use the algorithm Wu & Palmer(http://search.cpan.org/dist/WordNet-Similarity/lib/WordNet/Similarity/wup.pm) , using the implementation in [https://code.google.com/archive/p/ws4j/]. The consistency score for a record is then given as

$$\text{consistency} = 1 - \text{mean} \left( d(n_i, n_j) \right) , i = 1, \ldots, k, j = 1 \ldots, k, i \neq j$$

The comparative score is then determined as

$$\Delta_{\text{consistency}} = \text{consistency}_{\text{enriched}} - \text{consistency}_{\text{original}}$$

i.e., negative values indicate reduced consistency, positive values increased consistency. Slight reductions will be acceptable, while stronger reductions shall be used for indicating errors during enrichment.

**7.2.2 Representation of data quality measurements**

We need to represent the results of quality assessment in a well-defined and machine-readable way, as we aim at automatically comparing assessment results at different points in time (as the data being assessed is dynamic due to the structure of the EEXCESS system) and at different points in the workflow (e.g., input data quality with quality after enrichment).

We make use of the Data Quality Vocabulary (DQV) [DQV, 2015] currently under development at the W3C, which builds on DaQ [Debattista, 2014]. Note that this specification is only in a working draft stage, thus it does not yet cover all aspects that may be needed, but using the specification serves to validate the proposed model.
and develop it further. The model allows defining data sets and particular distributions of them (e.g., snapshots sampled at a certain point in time), as well as quality metrics. Quality measures, defined as specialisations of observations from DaQ, describe the result of applying a certain metric to a certain distribution of a data set.

An example DQV document with quality measurements is shown in Figure 18.

The quality analysis results can be visualised, e.g. as bar graphs. The inputs to this visualisation are the measurement results using DQV. The visualisation shown in Figure 19 is generated using an XSL transform, which creates an HTML page using jqPlot diagrams from the RDF/XML representation.

```xml
<rdf:RDF
  xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#
  xmlns:prov='http://www.w3.org/ns/prov#
  xmlns:dq='http://www.w3.org/ns/dataquality/
  xmlns:eexdaq='http://eexcess.eu/ns/dataquality/daq/
  xmlns:dqv='http://www.w3.org/ns/dqv#
  xmlns:dct='http://purl.org/dc/terms/
  xmlns:dcat='http://www.w3.org/ns/dcat#'
  xmlns:daq='http://purl.org/eis/vocab/daq#'>
  <daq:Metric rdf:about='eexdaq:metric#numberOfRecords'/>
  <daq:Metric rdf:about='eexdaq:metric#meanFieldsPerRecord'/>
  <daq:Metric rdf:about='eexdaq:metric#minFieldsPerRecord'/>
  <daq:Metric rdf:about='eexdaq:metric#maxFieldsPerRecord'/>
  <daq:Metric rdf:about='eexdaq:metric#meanNonEmptyFieldsPerRecord'/>
  <daq:Metric rdf:about='eexdaq:metric#meanNonEmptyFieldsPerDatafieldsPerRecord'/>
  <dcat:Dataset rdf:about='eexdaq:dataset#2015-10-14-17-30-26'>
    <dct:title>My EEXCESS dataset</dct:title>
    <dcat:distribution>
      <dcat:Distribution rdf:about='eexdaq:dataset#KIMCollectDistribution2015-10-14-17-30-26'>
        <dct:title>My EEXCESS KIMCollect dataset</dct:title>
        <prov:wasGeneratedBy rdf:resource='eexdaq:dataprovider#KIMCollect'/>
      </dcat:Distribution>
    </dcat:distribution>
  </dcat:Dataset>
  <dqv:QualityMeasure rdf:about='eexdaq:metric#numberOfRecordsKIMCollect2015-10-14-17-30-26'>
    <daq:value rdf:datatype='http://www.w3.org/2001/XMLSchema#double'>13</daq:value>
    <daq:computedOn rdf:resource='eexdaq:dataset#KIMCollectDistribution2015-10-14-17-30-26'/>
    <daq:metric rdf:resource='eexdaq:metric#numberOfRecords'/>
  </dqv:QualityMeasure>
  <dqv:QualityMeasure rdf:about='eexdaq:metric#meanFieldsPerRecordKIMCollect2015-10-14-17-30-26'>
    <daq:value rdf:datatype='http://www.w3.org/2001/XMLSchema#double'>33</daq:value>
    <daq:computedOn rdf:resource='eexdaq:dataset#KIMCollectDistribution2015-10-14-17-30-26'/>
    <daq:metric rdf:resource='eexdaq:metric#meanFieldsPerRecord'/>
  </dqv:QualityMeasure>
</rdf:RDF>
```

Figure 18: Data quality vocabulary example.
Figure 19: Data quality vocabulary visualisation
7.2.3 Implementation

We have built a prototype\textsuperscript{17} which uses data logged during calling the services from different data providers. Most services provide the data in XML. If the service provides the data in JSON, the data are transformed to XML by an internal service of the prototype. In particular, we analyse the input data and records, the number of returned fields per record, empty and non-empty fields. We also count mean links per record per provider and analyse which of the URIs in the dataset are reachable. As a result of the analysis the prototype generates statistics of the measured values and also generates charts.

The analysis is performed calling a Java application. Source can be found at Github repository\textsuperscript{18}. The application can either be run as JUnit test within Eclipse (testApplInputTestbedRandom100) or can also be exported as runnable JAR file e.g. named as dataquality-demo.jar and therefore be run as Java application from command line:

```
java -jar dataquality-demo.jar ./resources/input-testbed/random-100
```

It generates amongst others the mentioned bar graphs as well as csv-files and RDF/XML files containing information for further processing.

We have also implemented a web application and a JAVA desktop application which uses this library to analyse the metadata. The web application is deployed on http://eexcess-dev.joanneum.at/DataQualityWebApp/. Figure 20 shows the web application where the user can upload a XML file with the metadata. After uploading the file the user must define with a xpath-expression where the records are located in the uploaded XML file. Additional the user must define a name of this dataset, which will be used as label in the report and the charts. After pressing the button “analyse the data”, the web application uses the data quality lib to analyse the metadata. When this is finished the link to the generated report appears. This web application limits the size of the XML file to 200kB. When the user wants to process a larger file or multiple files, we provide a desktop application. The user can download this application from the web application. Figure 21 shows this application. Here the user has the same parameters to enter, like xpath to the records and name of the dataset. Instead of uploading one single file, the user here can specify a directory where the XML files are located. After pressing the button “analyse data” the application uses the data quality lib to process the metadata, as in the web application. When the processing is finished the report can open with the button “open report”.

\textsuperscript{17} Source code published at https://github.com/EEXCESS/data-quality

\textsuperscript{18} https://github.com/EEXCESS/data-quality
DataQuality

With this web-application you can analyse your data. For more general information about the EEXCESS project please visit http://eexcess.eu or https://github.com/EEXCESS.

For more information about this web app please visit https://github.com/EEXCESS/data-quality

File upload

Here you can upload your data as XML. The size of the file must be under 200kB. If you want to process more than one single file or bigger dataset we recommend to use the desktop application.

Choose

Please provide here the x-path where your records are located in your XML-structure.

X-Path:

/*[local-name()='BibliographicResourceCollection'][local-name()='BibliographicResource']

Name of the data provider:

The European Library

analyse the data

EEXCESS DataQuality report

After analysing your data you can reach the report here: DataQuality Report

EEXCESS DataQuality Desktop Application

If you want to process a larger dataset, we also provide a Desktop Application. To use this Desktop Application please download zip archive extract it and start the start.bat script.

Figure 20: DataQuality WebApp
7.2.4 Results

Record statistics

For testing our prototype we use a randomly selected subset data containing over 220,000 records from six data providers. Some data providers include only metadata fields in their service response, if a value for the actual object is present. That is the reason why we calculate the mean value of submitted metadata fields per
record for each data provider. The mean number of returned metadata fields varies between seven and 33 per record.

In Figure 22 we show the graph of data fields per record for each data provider. Figure 23 shows the mean non empty data fields per record for each provider in respect to their metadata fields overall. This visualises the filling degree of the record of a data provider.

![Figure 22: Input data quality - mean data fields/record](image-url)
Vocabulary quality

We also analysed the use of controlled vocabularies of the data providers involved. Most of the used vocabularies have been created by the data providers but are publicly accessible. Records from those data providers which use non-public vocabularies must be translated to other vocabularies during mapping.

We collect the number of terms identified by URIs, and analyse whether the URIs are actually URLs resolving to a definition of the resource. In addition, we analyse the types of the resources provided, i.e., whether there is only a human readable description or a machine readable description is provided.

Figure 24 shows the number of mean links per record per provider. The records from ZBW do contain references to controlled vocabularies, but not in the form of URIs, thus they have not been considered in this analysis. The data from DDB contained only few URIs.
Figure 24: Input data quality - number of mean links per record per provider.
During the enrichment process statistics of vocabulary servers are created. These statistics can be opened by e.g. Excel to make charts. The following charts are made in that way.

The diagram show in Figure 25 illustrates where the trusted vocabularies come from. This is an overview over the whole transformation process. That means that the links are counted before transformation, after transformation and after the enrichment. In our test set 16.6% of the vocabularies are from DBpedia followed by creativecommons.org (10.4%) and Europeana with 9.2%.
Figure 26: Vocabularies before transformation into EEXCESS data format

Figure 26 shows the vocabularies used in the original data files (before transform). Here DBpedia is on the first place with 12.3% followed by ddb.vocnet.org (10%) and creativecommons.org (7.9%).
Figure 27: Vocabularies after enrichment

Figure 27 shows that after enrichment the number of links for DBpedia increases significant to 26.4% also for Europeana and EEXCESS, whereas links to bibliographical sides are reduced. Furthermore the accessibility of the URLs used to identify vocabulary terms in the dataset is also checked. The following table shows that almost all URLs in the testbed-dataset are accessible. The number of URLs that are not accessible may vary from run to run of the analysis tool depending on the availability of the servers, although in most cases either an entire vocabulary is accessible or not.

Table 1: Total links vs. reachable links in dataset

<table>
<thead>
<tr>
<th>provider</th>
<th>#URLs</th>
<th>#accessible URLs</th>
<th>Difference (links not accessible)</th>
</tr>
</thead>
<tbody>
<tr>
<td>KIMCollect</td>
<td>1188</td>
<td>1186</td>
<td>2</td>
</tr>
<tr>
<td>ZBW</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Europeana</td>
<td>9905</td>
<td>9905</td>
<td>0</td>
</tr>
<tr>
<td>Wissenmedia</td>
<td>808</td>
<td>808</td>
<td>0</td>
</tr>
</tbody>
</table>
Regarding the figures for ZBW it is to be mentioned that the vocabularies that are used within ZBW are not linked to URIs that are publicly available at the moment. Further on an improvement could be to include URIs at ZBW’s side so that these data might be used for quality check.

Figure 28: Input data quality – Content Negotiation.

Figure 28 shows the possible variants of serving a resource for a DBpedia-URI. The desired information can be found in header-response with header name “Link”. This information is not always available e.g. for URIs from provider KIMCollect the server does not provide this information.

Regarding input data quality we also determine the hosts where the information of the possibilities a server may answer is available and where not.

7.3 Assessing Transformation and Enrichment Data Quality

7.3.1 Metrics and Implementation

For the assessment of the data quality we count the number of non-empty data fields during the enrichment process. That is before and after the transformation of the data files and after the enrichment. Additional we check the number of URIs during the enrichment. We use a blacklist to remove all unwanted administrative URIs.

7.3.2 Results

JR-DIG has developed a small set of tools, to create test data to evaluate the developed prototype. This Framework consists of several tools:

1) Tool to parse the logfiles and to extract the generated keywords. Further on to store them in a new file. The list of keywords is checked against duplicates. So the resulting list, is a list of all generated keywords, discounted how often the where used.

2) Tool using a list of keywords as input and querying all partner recommenders with these keywords. During the testbed the EEXCESS Framework had recorded the generated keywords. JR-DIG has taken a randomized set of 10 and 20 from those about 33000 keywords to build a Framework to test the enrichment services.
3) Tool to use these keywords to query the local deployed PartnerRecommenders, to generate the Files which our prototype then analyse to generate our data quality report.

Figure 29 shows the change of the number of data fields during the enrichment process. You can see different behaviours:

1. The amount of data fields of the DDB decrease significantly. The records of the DDB are quite large. Many of the data fields are not used by the EEXCESS transformation.

2. In general the number of data fields increases during enrichment. Especially by ZBW and Mendeley the increase is very significant. The reason is, that the enrichment process could find many additional terms in dbpedia and geonames.

Figure 29: data fields / record

In Figure 30 the number of URIs per records are shown. During transformation the amount of URIs decreases in Europeana und DDB. These data providers use very specific vocabularies, which are not used by EEXCESS. During the enrichment process the amount of URIs increase for all data providers.
This Section describes improvements and extensions of the mapping quality assessment task integrated in our metadata mapping configuration tool.

As presented in D4.3, assessing the mapping quality is integrated in our metadata mapping configuration tool. The underlying approach is based on performing a round trip mapping of a given metadata document and detecting differences in the original document and the corresponding mapping result. In this context, the term round trip mapping denotes that a metadata document is first mapped to an intermediate format and then mapped back to the original metadata format. Two different round trip mapping variants are implemented by the metadata mapping configuration tool. One uses only the internal conceptual representation of metadata properties of the configuration tool for performing a round trip mapping while the second employs a specific target metadata format. The first variant should be lossless in the normal case while in the second some loss of information or imprecision in the mappings may be expected based on the expressiveness of the formats involved.

After one of two available round trip variants are applied to a given metadata document, differences in the original document and the corresponding mapping result are observed. Such differences are detected by comparing the content of the involved documents addressed by XPath locators. In addition, designating an acceptable loss of information is enabled by different values of an XPath locator pair. In the initial approach, the specification of corresponding XPath locators has to be made by manually instantiating the comparison ontology (cf. D4.3) However, this approach has been improved. In the latest implementation, XPath locators are retrieved automatically by examining the structure of the involved metadata documents. Furthermore, an acceptable loss of information during the mapping process is stated by querying all possible mapping variants of the involved metadata elements.
For example, the quality of the mapping between two specific formats (KIM.Collect format and the Dublin Core Metadata Initiative\(^{19}\)) is assessed. The underlying mapping instructions are based on the defined mapping paths presented in Figure 31. The KIM.Collect metadata elements Autor and Hersteller are mapped to the conceptual metadata elements Author and Creator and vice versa. In addition, there is also a bidirectional mapping between the Dublin Core concept Creator and the conceptual metadata element Creator. When mapping between the specific formats, a loss of information has to be accepted. In fact, the KIM.Collect elements Autor and Fotograf map to the Dublin Core element Creator but this element maps to the KIM.Collect element Hersteller only. For demonstrating an unfinished mapping definition, the concept Fotograf is not involved in any mapping.

![Figure 31: Mapping paths between different metadata elements.](image)

In order to perform the two round trip mapping variants, these mappings are applied on the metadata document depicted in Figure 32. The mapping result of the round trip variant using the internal representation only is depicted in Figure 33. The XML structure of these documents is the same and also the concepts Autor and Hersteller are mapped without any loss of information. These concepts are represented by the tags <autor> and <hersteller> in the XML documents. However, there is currently no tag <fotograf> in the round trip mapping result available since the mapping has not been defined yet. When performing the mapping quality assessment this missing definition must be recognized.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<rtm_daten>
  <hauptebene>
    <fotograf>Peter Jackson</fotograf>
    <autor>Janine Potterbody</autor>
    <hersteller>Erik Dayhurst</hersteller>
  </hauptebene>
</rtm_daten>
```

**Figure 32: Example of an input metadata document.**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<rtm_daten>
  <hauptebene>
    <autor>Janine Potterbody</autor>
    <hersteller>Erik Dayhurst</hersteller>
  </hauptebene>
</rtm_daten>
```

**Figure 33: Example of a round trip mapping result via the conceptual representation only.**

\(^{19}\)http://dublincore.org/
The result of the round trip variant using a specific target format is depicted in Figure 34. Again, the XML structure has not been changed and the tag `<hersteller>` with the value Erik Dayhurst is available in both documents. Beside the missing tag `<fotograf>` (due to the uncomplete mapping definition), the tag `<autor>` is not present either. However, this is an acceptable loss of information based on the mapping defined between the KimPortal concept Autor and the Dublin Core concept Creator (cf. Figure 31). In any case, this variation must be reported during the quality assessment process.

```
<?xml version="1.0" encoding="UTF-8"?>
<rtm_daten>
  <hauptebene>
    <hersteller>Janine Potterbody</hersteller>
    <hersteller>Erik Dayhurst</hersteller>
  </hauptebene>
</rtm_daten>
```

Figure 34: Example of a round trip mapping result via specific target format.

Beside the automated selection of XPath locators for the assessment task, the presentation of the assessment results in the metadata mapping configuration tool has been improved as well. This assessment report is presented as a table, each content comparison step that is triggered by an XPath locator pair is represented by an own row. Clicking on these XPath locators enables a direct navigation to the corresponding locations in the XML documents (Input XPath and Output XPath). The values of these XPath locators are presented in separate columns (Input Value and Output Value). In addition, the assessment is result is explained as a short note (Result) and information about the origin of the specific XPath comparison is given (Origin). This information refers to location in the mapping creation process where the addressed XML tags are specified. The value of the origin information can either be “template”, meaning that the tags are specified as part of a mapping template, or “static” meaning that the tags are specified in the XML structure.

For example, in Figure 35 the assessment report of the round trip mapping variant via the conceptual representation only (cf. Figure 33) is depicted. Since the tag `<fotograf>` is missing after performing the round trip mapping, an error is reported in the first row of this report (Missing in output). Furthermore the existence of this tag in the input document is referenced and the value “template” in column Origin designates that a mapping template has to be fixed in order to solve the stated problem. As expected, no further errors are reported.

```
<table>
<thead>
<tr>
<th>Input XPath</th>
<th>Output XPath</th>
<th>Input Value</th>
<th>Output Value</th>
<th>Result</th>
<th>Origin</th>
</tr>
</thead>
<tbody>
<tr>
<td>/rtm_daten[1]/hauptebene[1]/fotograf[1]</td>
<td></td>
<td>Peter Jackson</td>
<td>Missing in output</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Figure 35: Mapping quality assessment report based of the round trip mapping variant via the conceptual representation only (cf. Figure 33).

In addition, Figure 36 depicts the assessment report of the round trip mapping variant via Dublin Core. Beside the missing `<fotograf>` tag, an accepted loss of information regarding the round trip mapping from the Kim Portal concept Autor to Hersteller via the Dublin Core concept Creator is reported. Finding the corresponding XPath locators, namely `/rtm_daten[1]/hauptebene[1]/autor[1]` and `/rtm_daten[1]/hauptebene[1]/hersteller[1]` is an automated process by considering the values of these locators in combination with the mapping instructions defined in the metadata mapping configuration tool such as concept mappings, data type representations, context information.
In order to hide the complexity of larger mapping quality assessment reports, certain assessment results can be regarded as approved and hidden (Button Approve). A simplified assessment report can be very useful when processing XML document containing many elements that are statically defined. In case the actual value of an approved assessment is in conflict with the expected result, the assessment result is reported anyway. Furthermore, the set of approved assessments can be retrieved and assessments can be removed from this approval list again (cf. Figure 37).

Figure 37: Example of approved locator associations.
8 Conclusions

With the PartnerWizard we have a new easier way to integrate new data provider into the EEXCESS framework. This work is not finished yet, we are working on the integration of the part for the query configuration from the KC. But we have already built several new PartnerRecommender with the PartnerWizard, which shows that this approach works well and we expect a higher quality if we have finished the integration with the query configuration.

Regarding the quality assessment we have added more and detailed information about the input data quality, we also analyse the data to detect structuredness in the input data. With our prototypes we have discovered several quality issues in the metadata of our data provider. We have informed our data provider about these issues and they are reviewing their metadata.
9 References


10 Glossary

Terms used within the EEXCESS project.

Partner Acronyms

- JR-DIG: JOANNEUM RESEARCH Forschungsgesellschaft mbH, AT
- Uni Passau: University of Passau, GE
- Know: Know-Center - Kompetenzzentrum für Wissenschaftsbasierte Anwendungen und Systeme Forschungs- und Entwicklungs Center GmbH, AT
- INSA: Institut National des Sciences Appliquées (INSA) de Lyon, FR
- ZBW: German National Library of Economics, GE
- BITM: BitMedia, AT
- KBL-AMBL: Kanton Basel Land, CH
- CT: Collection Trust, UK
- MEN: Mendeley Ltd., UK
- WM: wissenmedia, GE

Abbreviations

- EC: European Commission
- EEXCESS: Enhancing Europe’s eXchange in Cultural Educational and Scientific resource

Acknowledgement: The research leading to these results has received funding from the European Union’s Seventh Framework Programme (FP7/2007-2013) under grant agreement no 600601.