D2.1

Metadata benchmarking and curation module

Document information

Title	Metadata benchmarking and curation module
ID	CLARINPLUS-D2.1 (CE-2016-0742)
Author(s)	Davor Ostojić, Matej Ďurčo
Responsible WP leader	Dieter Van Uytvanck
Contractual Delivery Date	2016-03-31
Actual Delivery Date	2016-03-31
Distribution	Public
Document status in workplan	Deliverable

Project information

Project name	CLARIN-PLUS
Project number	676529
Call	H2020-INFRADEV-1-2015-1
Duration	2015-09-01 – 2017-08-31
Website	www.clarin.eu
Contact address	contact-clarinplus@clarin.eu

Table of contents

1 Executive Summary

This deliverable describes the metadata benchmarking and curation module developed as a component to be integrated in the CLARIN metadata infrastructure for curation, normalisation and quality assessment / benchmarking of single Metadata (MD) records, collections and Component Metadata (CMD) profiles. It is intended as technical support for the human curation work, aimed to improve the metadata quality in CLARIN (e.g. in the metadata curation taskforce).

The module can be applied in four different situations:

1. The metadata creator/editor needs to check (on-the-fly) the quality and validity of a newly created record.
2. The metadata modeler needs to evaluate the quality of profiles (especially the facet coverage), when selecting an existing or creating a new profile for a project / for new resources.
3. The data provider / repository admin / collection manager needs to check the overall quality of metadata in his/her repository, including the facet coverage.
4. All records ingested into the Virtual Language Observatory (VLO) have to undergo a systematic process of curation, validation, normalisation and quality assessment (benchmarking).

The primary output of the module is a detailed report in XML format, containing statistics and information about issues encountered during the validation and curation according to an array of quality criteria.

The module will be integrated with multiple components of the infrastructure:

- the VLO harvester and the whole VLO ingestion workflow
- the Component Registry and the CLARIN Concept Registry
- metadata authoring tools

For this, the module provides a RESTful API and can also be integrated into other components as a Java library. The curation reports can be used by other components in CLARIN's infrastructure for presentation purposes or further processing, or as a help for human users to identify quality issues and to address them.

2 Introduction

The curation module is a software component developed for the CLARIN metadata infrastructure for curation, normalisation and quality assessment / benchmarking of single MD records, collections and CMD profiles. The module is implemented in the Java[1] programming language. It can be used as stand-alone application, as a library in other components, as a (RESTful) web service and as a simple web application.

In the following chapters the main aspects of the curation module will be comprehensively described. In chapter 3, an overview of previously identified use cases and requirements, used as an input for design, is given. In chapter 4, the main new concepts and workflows are described. Chapter 5 describes the main components, packages and classes followed by UML diagrams. Chapter 6 details the structure of the reports (output) and how the score of the metadata quality is calculated, and lists the considered quality criteria. Chapter 7 gives instructions about installation and usage of the curation module. In the appendix readers can find some examples of generated reports using records from CLARIN's repository.

3 Use cases and requirements

In this chapter the list of use cases and requirements gathered by Metadata Curation Taskforce will be presented[2]. Use cases and requirements are taken over from CLARIN's wiki page[3][4].

3.1 Use cases

Use Case 1 – Metadata (MD) Creator checks the validity of newly created record

Purpose: Check validity of metadata record

Actor: MD Creator

Main Success Scenario:

1. User copies MD record into the web form and starts validation by clicking "Validate" button
2. Module executes schema validation, link check, vocabulary check, and facet coverage assessment
3. User gets the report with status, eventual errors and assessment
4. User gets instructions how to improve MD record (recommended profile, recommended values)

Use Case 2 – MD Modeler checks the quality of profiles

Purpose: Check quality of profile/schema

[1] https://docs.oracle.com/javase/7/docs/technotes/guides/language/

[2] https://trac.clarin.eu/wiki/Taskforces/Curation

[3] https://trac.clarin.eu/wiki/CurationModule

[4] https://trac.clarin.eu/wiki/Cmdi/QualityCriteria

Actor: MD Modeler

Main Success Scenario:

1. MD Modeler runs the curation module (ideally via a web interface) and passes as argument profile or schema
2. Module checks concept links and facet coverage
3. User gets the report on link status and facet coverage

Use Case 3 – Repository Administrator checks quality of metadata in his repository

Purpose: Check overall quality of metadata in repository

Actor: Repository Administrator / Collection manager

Main Success Scenario:

1. Administrator runs module (normally from command-line) and passes as argument location containing MD records
2. Module executes quality assessment of the records
3. Administrator gets summarized report on overall quality and individual issues of MD records in his repository

Use Case 4 – Curation Module in VLO workflow

Purpose: Curation Module in VLO workflow

Actor: VLO workflow

Main Success Scenario:

1. The next iteration of regular ingest to VLO is initiated by the system, starting with a fresh harvest by the harvesting module
2. When the harvest is finished the curation module is invoked to process the freshly harvested metadata (collection by collection)
3. Module executes validation, curation, (and normalisation) and generates reports that are stored next to the harvested metadata
4. It optionally generates normalised MD records
5. VLO importer can index the generated quality assessment metrics as additional information to the search engine SOLR[5]
6. It can optionally use also the normalised MD records for the indexing.
7. The reports, together with the statistics on the harvested records are available via a (administrative) web user interface (a VLO dashboard)

[5] http://lucene.apache.org/solr/

3.2 Requirements

- Given the huge number of records (currently 800K and continuously growing), curation module has to allow high throughput and should be scalable
- Ability to provide validation and / or assessment service for metadata creators and repository administrators (on instance level).
- CMD profile/schema assessment
- Process both single record and collections in batch mode
- Local and remote invocation
- Available via RESTful API and as a simple web interface for human interaction
- Evaluate following aspects: schema validation, URL inspection, value validation against vocabularies and normalisation (see the chapters 6.4 and 6.5 for the full list).
- Feedback about errors per record or per collection.
- Feedback about quality: score, facet coverage.
- Provision of instructions on MD improvement.
- Tracking of the quality over time for the analysis.

4 Concepts and Design

Main concepts (or classes) of the module are ***curation entity, processor, task*** and ***report***. Curation entity is an abstract object that represents something that can be curated. The requirements for this project specify three types of entities: CMD profiles, CMD instances and collections. Collection is understood as directory (potentially with subdirectories) containing CMD instances. Each curation entity has a specific type of processor. The curation process is divided into tasks, organised in a pipeline. Each task generates specific statistical information together with messages about issues that occurred during the processing and their severity level. Execution of the pipeline is stopped when one task generates messages with a fatal severity level. (This is necessary because some curation steps depend on previous ones, e.g. if a reference to profile or schema is missing, the mapping to facets cannot be determined). The information is collected in a report object. The report contains a different kind of information for each type of entity. At the end of the process, this report is serialized into an XML representation and printed or stored to disk, depending on the configuration.

The curation module accepts a path or a URL as main input parameter. The path can be a file or folder, which is automatically converted to a corresponding curation entity by the module. Based on the type of this entity, an adequate processor is invoked. In the following sections, the workflow for CMD instance and CMD collection is explained.

4.1 CMD Instance Curation Workflow

The workflow for CMD instances is presented in Figure 1.

CMD instance curation involves seven steps:

1. **File size** - The size of the instance file is compared with a given limit. This limit is configurable and is set to 10 MB by default in line with corresponding check in the VLO-importer. If the file size exceeds this limit, further processing is terminated.

2. **Process *CMD Header*** - check for presence of certain fields (*MdProfile*, *MdSelfLink* and *MdCollectionDisplayName*). The corresponding profile of the instance is identified either from the *MdProfile* field or from schema attribute. In case that the profile is undefined, execution is terminated.
3. **Process *ResourceProxies*** section - Information about the number of resource proxies, their types and MIME types are collected.
4. **XML validation** – The instance is validated against an XML schema (XSD) derived from the corresponding CMD profile. The XSD file is fetched from the CLARIN Component Registry and cached. Messages from the XML parser are collected, the number of complex, simple and empty XML elements is determined and all values with a URL are collected (based on string matching 'http.*').
5. **URL validation** - implemented by sending a HEAD request, collecting the HTTP response codes. (Note: This seriously degrades the performance, due to slow response time! This step can be optionally omitted, governed by input parameters.)
6. **Assess facet coverage** - values for VLO facets are extracted and facet coverage for the CMD instance and profile is calculated. For this purpose, facetConcepts.xml is used and the mapping is done in the same way as the VLO importer.
7. **Normalise facet values** – facet values are normalised, based on controlled vocabularies and normalisation maps.
   Note: This processing step is still in development, as it depends on restructuring the modularization of the VLO project, to allow reuse of the postprocessors from the VLO importer.

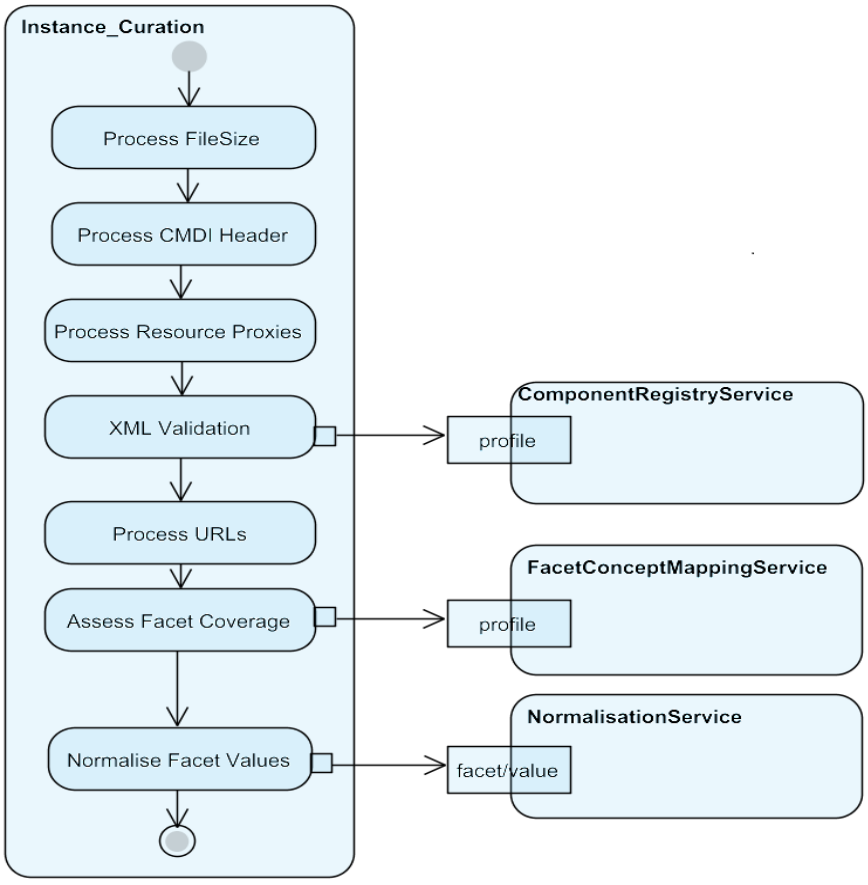

Figure 1: Instance Curation

4.2 Collection Curation Workflow

Starting from the root directory, the directory tree is visited in post-order and after visiting each directory all of its children (subdirectories and files) are processed in parallel, allowing substantial performance gains. When the report for each child is generated (instance or subdirectory), all children's reports are recursively aggregated into a collection report of the parent.

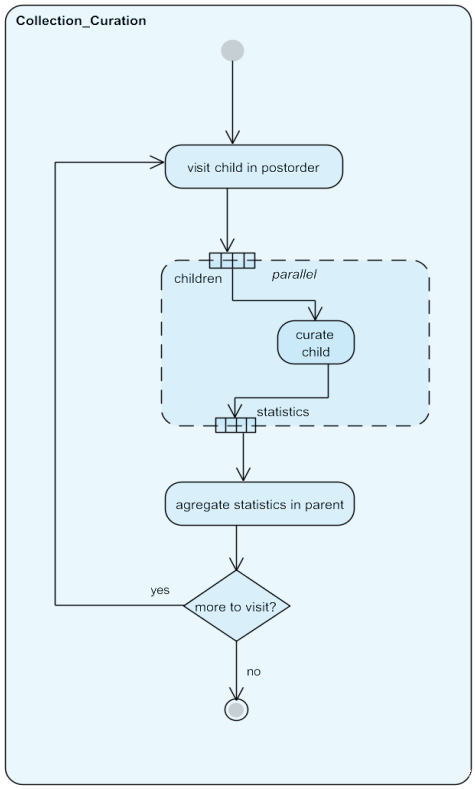

Figure 2: Collection Curation

5 Components and Implementation Details

The curation module is developed using Eclipse Mars[6] as IDE, with **Java** version 1.8.0_51 and Apache Maven 3.3.3. The RESTful API is implemented using the Jersey 1.9 library. For version control system git[7] and the corresponding plugin for eclipse egit 4.1 are used. The code is available at GitHub via: **https://github.com/acdh-oeaw/clarin-curation-module.git**.

The curation module doesn't have explicit dependencies on other CLARIN projects but some of the ideas and designs are taken from the following software projects[8]:

- CMDIValidator
- VLO

[6] https://eclipse.org/mars/

[7] https://git-scm.com/

[8] https://trac.clarin.eu/wiki/SoftwareDevelopment

From the VLO project, the logic of a mapping to facets is taken and modified to serve concurrent requests. In the future, when the normalisation service is implemented, the VLO-vocabulary project will be used.

As described in the previous chapters the main entities are the curation entity, processor, task and report. Figure 3 shows the class diagram with corresponding classes for these 4 entities.

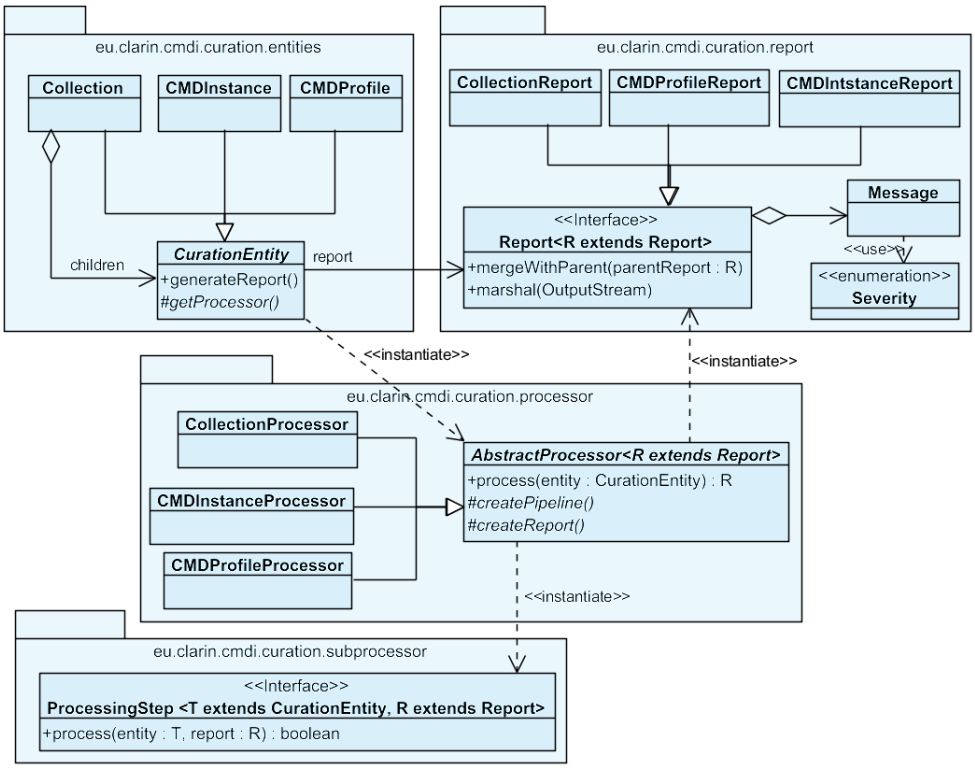

Figure 3: Curation Class Diagram

The class diagram presents only the most important classes and relations. When the curation is started adequately, a *CurationEntity* object is instantiated. If the user passes a path as input parameter, the type of the curation entity is resolved automatically. In case of file the type is resolved according to the file extension, in case of .xml *CMDInstance* is created while in case of .xsd *CMDProfile*. In the future this has to be changed in order to support other types like different xml formats. After instantiation of the *CurationEntity,* the method *generateReport* is called. This method creates an instance of the *AbstractProcessor* specific for the curation type and then calls its *process* method.

AbstractProcessor contains the method *createPipeline* which returns a sequence of *ProcessingStep* objects. The package *eu.clarin.cmdi.curation.subprocessor* contains also a number of implementations of the *ProcessingStep* interface where each class represents one step from the curation workflow. The method *process* calls in sequence each of these objects and at the end it returns the report. The interface *CurationTask* is generic and during implementation the developer needs to specify the *CurationEntity* type and *Report* type. This means that one implementation of this interface can work only with a specific entity, which limits the reusability of its code. The reason for this design was the fact that 3 starting entities are very different from each other and there is little overlap in the tasks and the report format.

The interface *Report* defines two methods: *mergeWithParent* and *marshal*. This interface is also generic and the user has to specify the type of the parent collection. With the method *mergeWithParent*, a class defines how the statistical information will be merged into the parent's

report. This way the responsibility of the report aggregation is transferred to the children because the parent can have different types of children, for example a directory can contain another directory or files. The second method, *marshal* is used for serialisation. Currently only the XML format is supported, using JAXB. The current design allows only one way of serialisation, in case that multiple formats are required in the future this part has to be redesigned, possibly using the Strategy pattern[9].

Other components of the project will be presented below. Among them, the two most important ones are *ComponentRegistryService* and *FacetConceptMapping*.

5.1 *ComponentRegistryService* Component

Since the CMD profile definitions and thus the derived XML Schemas are immutable[10], they can be safely cached locally, yielding a significant performance gain. The *ComponentRegistryService* component acts as an internal proxy (featuring caching functionality) to the Component Registry. In Figure 4 the workflow of this component is given.

The component keeps the XML schemas as an object in an in-memory cache. On request, if a schema is not in the cache, the component searches for it on local disk in a location specified in the configuration. If it is not on the disk, it will be downloaded from CLARIN's Component Registry using its RESTful API and stored to local disk. When the schema is loaded, the component parses it and puts it in the in-memory cache. To increase performance this component is designed to serve concurrent requests. During run time there is only one instance of this component (singleton).

This component is implemented in the class eu.clarin.cmdi.curation.component_registry. *ComponentRegistryService*. The most important methods of this class are *getPublicProfiles* and *GetSchema*. The first method returns all public profiles from component registry while the second maps the profile to java's internal representation of XML schema - javax.xml.validation.Schema.

5.2 FacetConceptMapping Component

The *FacetConceptMapping* component does the mapping from profile to facets by using facetConcepts.xml. This file will be downloaded from CLARIN's VLO code repository[11].

The component is also able to serve concurrent requests and it uses the Singleton pattern. The implementation with ancillary classes can be found in *eu.clarin.cmdi.curation.facets* package. The component maintains the mapping of profile to the facets with corresponding XPaths. It is used by the *InstanceFacetHandler* task to obtain the XPaths to extract values for facets. The workflow is given in the Figure 5. The mappings (XPaths) are generated combining information from the schema (provided by *ComponentRegistryService*) and the *facetConcepts.xml* configuration file.

[9] See https://en.wikipedia.org/wiki/Strategy_pattern

[10] except unpublished ones

[11] https://raw.githubusercontent.com/clarin-eric/VLO/master/vlo-commons/src/main/resources/facetConcepts.xml.

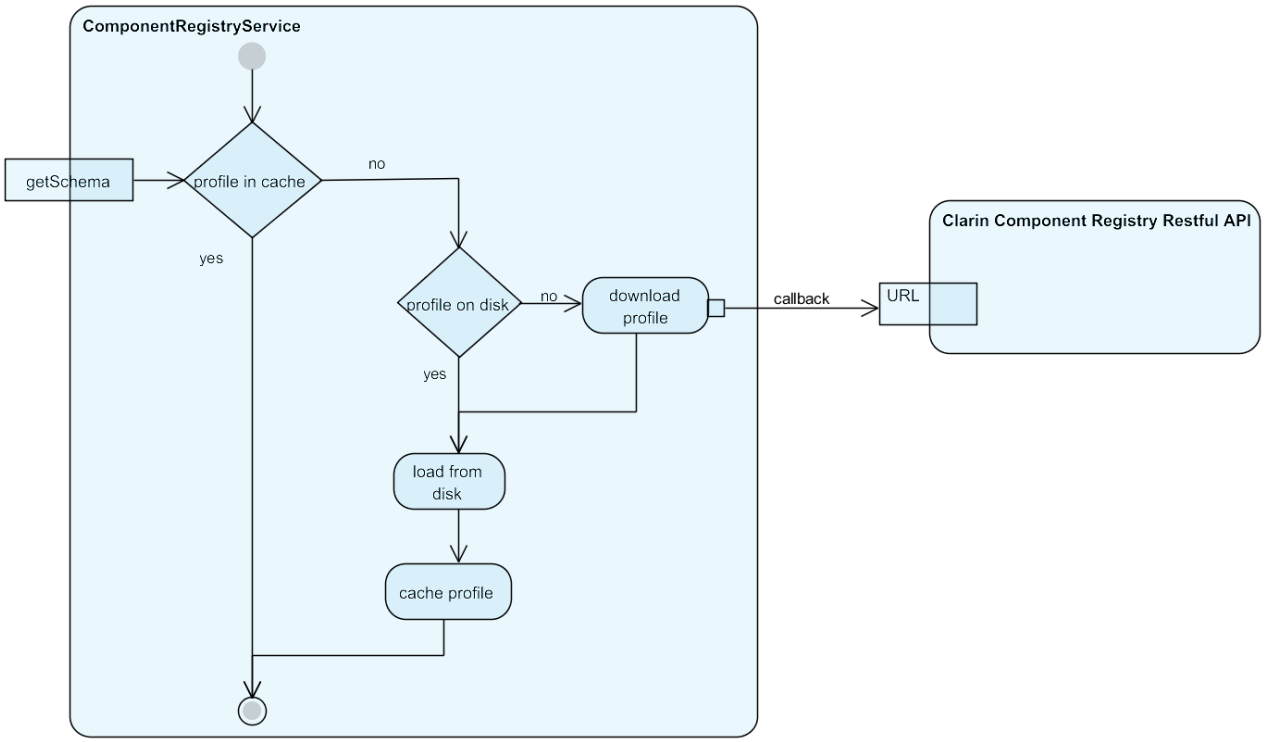

Figure 4: Component Registry Workflow

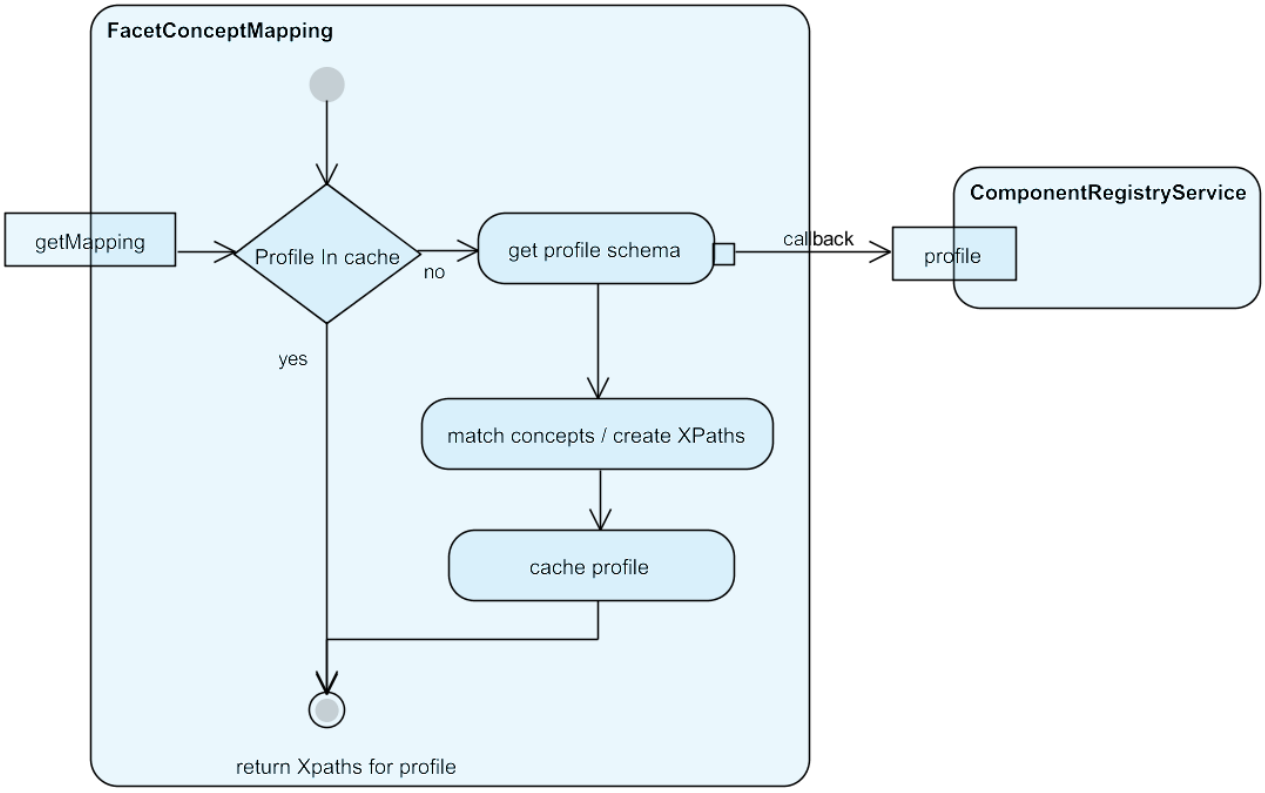

Figure 5: Facet Concept Mapping

6 Reports and Scoring

This chapter presents the structure of the three report types. At first quality criteria as agreed upon by the CLARIN technical team is listed, then the structure and fields of the reports are explained and finally the score calculation is explained.

6.1 Report for profile curation

In the profile curation, the curation module generates a report containing following information:

- **timestamp** – the timestamp in format YYYY.MM.DD.HH.MM.SS when a report is created. It can be used for time tracking of the report
- **score** – total score of the profile in format score/max score. Details can be found later in this chapter
- **ID** – the id of the profile as issued by Component Registry
- **name** – name of the profile
- **description** – description of the profile
- **isPublic** – tells if the profile is public
- **components**[12] – a section listing CMD components listed in the profile. It contains information about the total number of components and number of required components. The component is required when it has attribute *minOccurrence* greater than 0
- **component** – represents a single component from the **components** list. It contains following information:
  - **name** – name of the component
  - **id** – id of the component
  - **required** – tells if component is required
- **numOfElements** – number of the xml elements with name "*element*"
- **numOfRequiredElements** – number of elements having the attribute *minOccurrence* greater than 0
- **ratioOfElementsWithDatcat** – percentage of the elements having specified data-category attribute.
- **concepts –** a list of concepts (defined by CCR (CLARIN Concept Registry)[13]) with following aggregated information
  - **total** – total number if data-categories in the profiles XSD
  - **unique** – number of unique data-categories
- **concept –** listed data-category with its url and number of occurrences in a profile
- **facet-section** –information about facet coverage. It includes information about the total number of facet (comes from facetConcepts.xml) and subsection **profile**. This subsection has following fields:
  - **numOfCoveredFacets** – number of facets covered by this profile
  - **coverage** – percent of covered facets
  - **not-covered** - list of facets not covered by this profile

[12] The term 'component' is used here in the sense defined by the CMDI, as a section of a metadata profile which shares a certain characteristics of properties.

[13] https://openskos.meertens.knaw.nl/ccr/browser/index.php

6.2 Report for instance curation

In the instance curation, the curation module generates a report containing following information:

- **timestamp** – the timestamp in format YYYY.MM.DD.HH.MM.SS when the analysis was performed and the report is created.
- **score** – total score of the instance in format score/max score. Details can be found in chapter 6.8
- **isValid** – tells if the processing was interrupted due to the unrecoverable error during curation process. In case when report is invalid the score is 0
- **file-section** –information about instance file like full **path** and **size** in bytes
- **header-section** –information from the *Header* section of CMD instance:
  - profile – instances profile
- **resProxy-section –** information from the *ResourceProxyList* section of CMD instance:
  - **numOfResProxies** – total number of *ResourceProxy* elements
  - **numOfResWithMime** – total number of resources with specified MIME type
  - **percOfResProxiesWithMime** – percentage of resources with specified MIME type **numOfLandingPages** – total number of resources of type "LandingPage"
  - **numOfResProxiesWithReferences** – total number of resources followed by reference to the actual resource
  - **percOfResProxiesWithReferences** – percentage of resources followed by reference to the actual resource
  - **resourceTypes** – a list of values extracted from resourceType field with number of occurences
- **xml-validation-section** - information collected during validation against profiles XSD:
  - **numOfXMLElements** – total number of XML elements in instance's XML file
  - **numOfXMLSimpleElements** – total number of elements that can carry a value (simple element)
  - **numOfXMLEmptyElement** – number of simple elements with missing value
  - **percOfPopulatedElements** – percentage of simple elements with specified value
- **url-validation-section –** information collected during link validation:
  - **numOfLinks** – total number of URLs in the element values
  - **numOfUniqueLinks** – number of unique links
  - **numOfResProxiesLinks** – number of links coming from *ResourceProxies* section
  - **numOfBrokenLinks** – number of broken links (status > 400)
  - **percOfValidLinks** – percentage of valid (unique) links
- **facets** – section about facet coverage and facet values it has information about total number of facets (**numOfFacets**) and two subsections **profile** and **instance**.
- **profile** – section about facet coverage of the instance's profile with following information:
  - **numOfCoveredFacets** – number of facets covered by this profile
  - **coverage** – percentage of covered facets
  - **not-covered** – list of facets names not covered by the profile
- **instance** – section about facet coverage on the instance level and facet values extracted from the instance**:**
  - **numOfCoveredFacets** - number of facets with specified value
  - **coverage** - coverage of the instance
  - **values –** list of the extracted values for each facet (facets can have more than one value)
  - **missingValues** – list of facets without specified value

Each of the main sections has a list of details, messages about warnings and errors that occurred during the curation process. This section is visible only if it contains at least one message. The format of the message is

- **lvl** – severity of the message. "WARNNING", "ERROR" and "FATAL"
- **message** – description of the error

6.3 Report for collection curation

In the collection curation, the curation module generates a report by aggregating information from its children (instance or another collection). It contains following information:

- **timestamp** – the timestamp in format YYYY.MM.DD.HH.MM.SS when the analysis was performed and the report created.
- **score** – total score of the instance in format "sum of scores of all items/max score for collection". Details on how the score is computed can be found in 6.6
- **avgScore** – average score among all instances in the collection in the format "average score/max score for one instance"
- **file-section** – contains general information about files in collection:
  - **provider** – name of the collection (directory name)
  - **numOfFiles** – number of instances in the collection
  - **size** – total size in bytes of the collection
  - **avgSize** – average instance size in bytes
  - **minFileSize** – size of the smallest file in the collection
  - **maxFileSize** – size of the smallest file in the collection
- **header-section** – contains the list of used profiles and their counts as well as count of unique profiles in collection
- **resProxy-section** – contains aggregated statistics for total and average values from resProxy-section of the instances. For details see this section
- **xml-validation-section** – contains aggregated statistics for total and average values from xml-validation-section of the instances. For details see this section
- **xml-validation-section -** contains aggregated statistics for total and average values from url-validation-section of the instances. For details see this section
- **facet-section** – contains information about average facet coverage for instances in the collection
- **invalidFilesList** – list of invalid CMD records, present if at least one is invalid.

6.4 Quality Criteria for schema/profile

In the following we list the quality criteria against which the CMD profiles are evaluated[14]. They are divided into strict and soft criteria. Soft criteria are rather of informative nature, strict criteria imply a quality issue of certain severity level.

Strict criteria:

- ratio of elements with concept link
- presence of "required"[15] concept links

[14] https://trac.clarin.eu/wiki/Cmdi/QualityCriteria

[15] for now "required" concept links are those that go into VLO facets, adjustments to the list need further discussion

- coverage of mapping to VLO facets (possibly weighted by facets)

Soft criteria:

- public accessibility of the profile
- status of the concept (approved/candidate/deprecated/...)
- number of defined elements (identify a range - too many, too little)
- number of unique components used
- total number of components used
- number of references to distinct data categories defined externally

6.5 Quality Criteria for instance

In the following we list the quality criteria for individual CMD record

Strict criteria:

- availability of the (reference to) schema (i.e. there is a reference to schema and the schema can be retrieved)
- the availability of schema in the Component Registry
- validity of the record with regard to the schema
- completeness of the header fields, availability of
    - unique *MdSelfLink*
    - *MdCollectionDisplayName*
    - *MdProfile*
- *ResourceProxy* elements
    - availability of LandingPage
    - availability of mime type
    - availability *ResourceProxy* (with *@type=Resource* or *@type=Metadata* available)
- URL-inspection - broken links
    - in *MdSelfLink*
    - in *ResourceProxy*
    - in the CMD_elements (less important)
- filled-in ratio / sparseness
    - how many of the elements defined by schema are actually populated with information
    - completeness concerning relative importance
    - coverage of the VLO facets (instance level) weighted per facet
- size
    - VLO maximum file size limit
    - max number of *ResourceProxies* per record

Soft criteria:

- values conform to a controlled vocabularies (where applicable)
- number of elements
- length of the strings in description fields
- information entropy (a lot of very similar files might be an indication of a suboptimal modelling)

There are a number of further aspects, metrics that could be evaluated, however these would need more clarification, and are currently excluded:

- publication before creation date
- editing quality: typos, expected format(date, capitalization, plural vs. singular)
- the combination of values in a related categorical fields is not consistent (if field A has value x, field B should have y or z)
- accuracy – semantic distance between data and MD
- conformance to expectation metrics
- coherence – different metadata fields describe the same object in the same way
- accessibility metrics - retrieval and cognitive? check if you can retrieve the data, or there is login (and Federated Login[16])
- longitudinal metrics - how does the quality change over time?

6.6 Score Calculation and Benchmarking

In this chapter we explain the calculation of the metadata quality score. Note that the score should be regarded as an indicator, and cannot be interpreted without context. All scores are presented as a ratio "score"/"max score" to give an insight into "how good is the quality". Higher score means better quality. The score is calculated as the sum of the points given by the individual tasks. The current implementation considers only strict criteria for score calculation. In the initial phase, each condition gives 1 or 0 depending on if it is fulfilled or not, but in the future the scoring must be weighted or calibrated. This calibration requires enough assessed material and the assessments require manual evaluation, as to be able to decide on the relative importance of individual criteria. The already assessed data will then serve as a **benchmark**, against which new data can be assessed. Here we will follow the workflow proposed by Kemps-Snijders [2].

In the next section, the score calculation for each of the three types of curation is explained in detail.

6.7 Profile Score

Score for the profile is influenced by the following:

- is the profile public (1 if yes, 0 otherwise)
- percentage of the elements with specified data-categories (value in range 0-1)
- facet coverage (value in range 0-1)

Thus the total scale ranges from 0 to 3.

6.8 Instance's Score

Score for the instance is influenced by the following:

- size of the file (less than 10MB by default 1, 0 otherwise)
- availability of schema (1 or 0)
- schema resides in CLARIN's Component Registry (1 or 0)
- *mdProfile* value in header exists (1 if value exists in header, 0 otherwise)
- *mdCollectionDisplay* (1 if value exists in header, 0 otherwise)

[16] http://clarin.eu/content/federated-identity

- *mdSelfLink* (1 if value exists in header, 0 otherwise)
- resource proxies (1 if at least one exists, 0 otherwise)
- percent of resource proxies having link (value in range 0-1)
- percent of non-empty xml elements (value in range 0-1)
- percent of valid links (value in range 0-1)
- facet coverage (value in range 0-1)

Score is calculated by summarizing points, and the total scale range will be 0 to 11.

6.9 Collection Score

Collection has two types of score, total and average. The total score is calculated by summarizing scores for each CMD record in collection while the average score refers to the average CMD instance score.

7 Installation and Usage

7.1 Getting the code

The code for this project is available from GitHub repository under the name clarin-curation-module. URL of the project is: https://github.com/acdh-oeatw/clarin-curation-module.git.

User can either use git command line interface or graphical interface to fetch the library or to download the library manually from GitHub. If git is installed the user can run the following command:

git clone https://github.com/acdh-oeaw/clarin-curation-module.git

7.2 Installation

Once the code is downloaded the user needs to build an executable version. To create a distribution maven[17] (2 or greater) is required. When maven is installed one can run the following command from projects directory:

mvn clean package

In case of success this command should produce an archive in a target folder containing an executable jar file, configuration file, and starting script.

7.3 Configuration and running

After unpacking the archive the user can call the starting script. To run the application, java version 8 is required. The application accepts three parameters from command line:

- -i path1 path2 … or –i path1 –i path2
- -c path_to_configuration_file (optional)
- -p (optional)

[17] https://maven.apache.org/

-i is used to pass the path to the files or folder for curation. –c is optional and is used to pass the path to the configuration file. If it is not specified, the application will use the default one in its folder. The last parameter –p is used when the user wants to print the result on the screen instead of saving it in the file.

The configuration file is in XML format and contains following options:

- MAX_SIZE_OF_FILE
- HTTP_VALIDATION
- GENERATE_CHILDREN_REPORTS
- OUTPUT_DIRECTORY

MAX_SIZE_OF_FILE value represents the maximal size of the file in kilobytes that can be processed. In case that the file size exceeds this value, the file will be considered as invalid and removed from further processing. Default value is 30 KB which originally comes from VLO restriction.

With HTTP_VALIDATION option the user chooses to include link validation step in validation pipeline. Allowed values are “true” and “false”, default is “false”. Because of the huge impact on performance one should use this option for single records or smaller collections, less than 1000.

When GENERATE_CHILDREN_REPORTS sets, the application will save reports from each child (file or folder) on disk, otherwise only one, aggregated on the top level, report is saved.

OUTPUT_DIRECTORY represents the path where the report will be saved. The directory must exist beforehand. The default value is “./reports”.

8 VLO integration

The Curation module shall be employed primarily within the ingestion workflow of the VLO. In the processing pipeline it shall process the output of the harvesting module and feed into the VLO-importer. There are multiple options for the interaction between the curation and the importer. One is that the curation module creates a normalised version of the CMD records still adhering to the original schema/profile. However the produced instance reports also feature a separate section with facet values (see appendix 11.1 for details). This section could be consumed by the VLO-importer for transformation to SOLR documents but this option requires discussion with the VLO development team and modifications of the report format.

The module itself has no persistence layer and all results need to be further processed by the invoking system. The plan is to store the reports next to the individual time-stamped harvest datasets. In this way the metadata and the corresponding reports are co-located and can easily be used for further processing. This is especially interesting for the planned comparison of the quality and overall development of the datasets over time.

9 Next steps

The base layer for the curation process has now been established. As a next step it will be tightly integrated into the harvesting and VLO-ingestion workflow. Thereby it will be connected to a dashboard application that allows to overview and manage all the steps of the workflow. This process is depicted in section 11.4, illustrating the position of the curation module within the bigger picture.

On the short-term, refinement of the quality score metric and a normalisation step for facet values will be taken care of. One of the top priorities is to include the score of the profile into instance's score. Another important point is to include weights in score calculation.

For presentation purposes an XSLT needs to be provided in order to transform the XML format of report to a human-readable HTML format. Optionally, the curation module can be extended to accept report as another type of input for aggregation of already assessed records. In the mid-term, the curation module needs to be integrated into the overall VLO ingestion workflow, especially to be coupled with the harvester reimplementation and the envisaged VLO-dashboard

10 Conclusion

The curation module can be used for curation and quality assessment of CMD profiles, instances and collections. Reports produced by the module help data providers and metadata curators to get an assessment of the overall quality of metadata in a collection and to identify issues problems the reasons for quality score. They contain statistical information for consumption by other CLARIN software components and human messages to help curators to improve metadata quality. Quality assessment for profile and instance can be done from a web browser and/or via a RESTful API. This enables metadata authors to do a real time validation and curation over the network to immediately get suggestions of how to improve the quality before data is exposed.

11 Appendix

11.1 Example Profile Report

Example of report generated for profile AnnotatedCorpusProfile-DLU:


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<profile-report>
   <timeStamp>2016.03.17.15.17.27</timeStamp>
   <score>2.527/3.000</score>
   <ID>p_1361876010587</ID>
   <name>AnnotatedCorpusProfile-DLU</name>
   <description>A CMDI profile for annotated text corpus resources.</description>
   <isPublic>true</isPublic>
   <cmdi-components-section>
      <total>57</total>
      <unique>31</unique>
      <required>25</required>
   </cmdi-components-section>
   <cmd-elements-section>
      <total>197</total>
      <unique>111</unique>
      <required>86</required>
      <withDatacategory>119</withDatacategory>
      <percWithDatacategory>0.6040609137055838</percWithDatacategory>
      <concepts total="119" unique="58" required="55">
         <concept url="http://hdl.handle.net/11459/CCR_C-3806_e55e9ed6-b099-c21d-a634-3c7f4d22a215" count="1"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-2969_b04052d7-f726-ae8d-84f3-f9838e7bc296" count="1"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-2453_1f0c3ea5-7966-ae11-d3c6-448424d4e6e8" count="1"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-2571_2be2e583-e5af-34c2-3673-93359ec1f7df" count="1"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-2092_36cd7ca8-e412-9f29-7ea7-4a3ba4ba2c91" count="2"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-3812_ce466764-27ab-b97b-a0bb-d7ac23ad9e60" count="5"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-2457_45bbaa1a-7002-2ecd-ab9d-57a189f694a6" count="1"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-2522_3bdc6af1-bf1b-3f5d-2938-62d99a1980ab" count="1"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-2967_c96c9be1-3655-28c8-2b1c-0c499309c87a" count="1"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-2490_44bc38a3-1799-4149-c791-40ac0176f0ff" count="1"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-3823_21273bbe-3d22-38cd-9a9c-85cc8807d087" count="1"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-2468_e4135e12-c272-171e-a8a2-48339228387b" count="1"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-2979_8030473e-bbcb-6b87-3fd2-90554429ec50" count="5"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-2482_08eded24-4086-7e3f-88e5-e0807fb01e17" count="2"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-2564_880b1108-6b03-647f-eed9-cdfbd464c661" count="1"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-3791_8f409aa8-2d57-5fc6-23f0-f490471807d8" count="1"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-3807_1479532e-b741-f5d4-99f4-881c5908cfd9" count="3"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-2527_77dad635-e029-acca-2fa8-f2fa5751d3af" count="1"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-3809_9b539a31-f06a-1a0c-6592-f51f366df740" count="1"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-2535_a4f0e587-455a-ef40-c129-9f605e4c8fad" count="1"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-2514_fd520a58-838b-9b79-e420-682e07cfa702" count="1"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-2485_474ece8d-abc9-90af-8c8a-5df540a0f970" count="1"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-4119_ad3a9349-db08-cefd-2c5e-0f311f3f0b54" count="1"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-3792_68c770a4-d58c-46dd-d429-5609ce5f81c3" count="2"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-2978_0e9e4864-44c4-de22-66b1-9b38bca10836" count="3"/>
         <concept url="http://purl.org/dc/terms/creator" count="1"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-2502_747eb0cd-03e9-cffb-34cc-d0c8c77e4c5a" count="1"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-2526_979ac535-eaa5-5e59-3cad-51c450234698" count="1"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-2536_13fc5f10-c14a-1f64-a669-32736f6d3ef5" count="1"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-2509_3b86afe2-ebde-ba09-8a1c-fe6bdc46a739" count="2"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-2520_9eeedfb4-47d3-ddee-cfcb-99ac634bf1db" count="24"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-3818_8c4aec73-1654-7565-9575-c4a17425ee29" count="1"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-3793_3bbfdf57-c9c8-c45b-f1c0-52773b1b9dbc" count="1"/>
         <concept url="http://hdl.handle.net/11459/CCR_C-2494_2451c60f-fd9f-6c36-02f6-ac5b8929f487" count="1"/>
```

```
        <concept url="http://hdl.handle.net/11459/CCR_C-2538_8b697452-7ef3-9fce-ccf9-a7f344f11317" count="2"/>
        <concept url="http://hdl.handle.net/11459/CCR_C-2544_3626545e-a21d-058c-ebfd-241c0464e7e5" count="1"/>
        <concept url="http://hdl.handle.net/11459/CCR_C-3822_ed57a8f6-05f2-0731-6350-8158e74fcb5f" count="1"/>
        <concept url="http://hdl.handle.net/11459/CCR_C-2537_fa206273-223a-f4fa-dde3-ba59b965701f" count="1"/>
        <concept url="http://hdl.handle.net/11459/CCR_C-2484_669684e7-cb9e-ea96-59cb-a25fe89b9b9d" count="2"/>
        <concept url="http://hdl.handle.net/11459/CCR_C-2505_b61e249b-ac68-b40a-0f21-03a4a26e16b4" count="5"/>
        <concept url="http://hdl.handle.net/11459/CCR_C-2455_56998f5b-47bc-3d58-3bf6-13d18d2b0045" count="3"/>
        <concept url="http://hdl.handle.net/11459/CCR_C-2539_f831f74e-f8ca-4e29-bb02-eb6ca7ea3073" count="2"/>
        <concept url="http://hdl.handle.net/11459/CCR_C-2499_52993a80-0bcf-d671-22dc-903effdb98b7" count="1"/>
        <concept url="http://hdl.handle.net/11459/CCR_C-2462_5b266792-7a54-9296-42e2-2318b9861fe1" count="1"/>
        <concept url="http://hdl.handle.net/11459/CCR_C-2546_180dca37-c1d8-dffe-5d46-8f16de143320" count="8"/>
        <concept url="http://hdl.handle.net/11459/CCR_C-3810_7d3b783a-d2c6-51fc-01ab-6a8a658d94c8" count="1"/>
        <concept url="http://hdl.handle.net/11459/CCR_C-2545_d873f2ab-2a2f-29d6-a9ab-260cde57f227" count="1"/>
        <concept url="http://hdl.handle.net/11459/CCR_C-2467_f4e7331f-b930-fc42-eeea-05e383cfaa78" count="1"/>
        <concept url="http://hdl.handle.net/11459/CCR_C-2547_7883d382-b3ce-8ab4-7052-0138525a8ba1" count="2"/>
        <concept url="http://hdl.handle.net/11459/CCR_C-5616_c39f4629-3f14-8ceb-f613-0610d30a5ff0" count="1"/>
        <concept url="http://hdl.handle.net/11459/CCR_C-2506_48f68696-57f3-74da-38e8-aa0f8e6ecc2f" count="1"/>
        <concept url="http://hdl.handle.net/11459/CCR_C-2491_e2d90ef0-a2e9-c101-6d35-bf25fc29f901" count="1"/>
        <concept url="http://hdl.handle.net/11459/CCR_C-2956_519a4aab-2f76-0fd3-090e-f0d6b81a7dbb" count="1"/>
        <concept url="http://hdl.handle.net/11459/CCR_C-2461_76a92748-423e-46a0-cc7c-84a7d1c46794" count="3"/>
        <concept url="http://purl.org/dc/terms/title" count="1"/>
        <concept url="http://hdl.handle.net/11459/CCR_C-2521_7b01b455-0de8-d753-ad4e-dee49953ae98" count="3"/>
        <concept url="http://hdl.handle.net/11459/CCR_C-3814_dce1ba90-46f9-3b4a-d1ce-b37173126473" count="2"/>
        <concept url="http://hdl.handle.net/11459/CCR_C-3801_99745da2-854a-9b12-a92d-ce24f05c8bb3" count="1"/>
      </concepts>
    </cmd-elements-section>
    <facets-section>
      <numOfFacets>26</numOfFacets>
      <profile>
        <numOfCoveredFacets>24</numOfCoveredFacets>
        <coverage>0.9230769230769231</coverage>
        <not-covered>
          <facet>distributionType</facet>
          <facet>rightsHolder</facet>
        </not-covered>
      </profile>
    </facets-section>
</profile-report>
```


11.2 Example Instance Report

Example of the report generated for instance Instituut_voor_Nederlandse_Lexicologie_INL_Metadata_Repository /eeba6672493adaa6f24b3bf1c13c385b.xml


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<instance-report>
    <timeStamp>2016.03.17.17.34.41</timeStamp>
    <score>8.807/11.000</score>
    <isValid>true</isValid>
    <header-section>
      <profile>p_1271859438164</profile>
      <details>
        <messages lvl="ERROR" message="CMDI Record must contain CMD/Header/MdProfile tag with profile ID!"/>
      </details>
    </header-section>
    <file-section>

<path>D:\data\cmdi\Instituut_voor_Nederlandse_Lexicologie_INL_Metadata_Repository\eeba6672493adaa6f24b3bf1c13c385b.x
ml</path>
      <size>7433</size>
```

```
</file-section>
<resProxy-section>
  <numOfResProxies>3</numOfResProxies>
  <numOfResProxiesWithMime>1</numOfResProxiesWithMime>
  <percOfResProxiesWithMime>0.3333333333333333</percOfResProxiesWithMime>
  <numOfResProxiesWithReferences>3</numOfResProxiesWithReferences>
  <percOfResProxiesWithReferences>1.0</percOfResProxiesWithReferences>
  <resourceTypes>
    <resourceType type="SearchService" count="1"/>
    <resourceType type="Resource" count="1"/>
    <resourceType type="LandingPage" count="1"/>
  </resourceTypes>
</resProxy-section>
<xml-validation-section>
  <numOfXMLElements>96</numOfXMLElements>
  <numOfXMLSimpleElements>64</numOfXMLSimpleElements>
  <numOfXMLEmptyElement>14</numOfXMLEmptyElement>
  <percOfPopulatedElements>0.78125</percOfPopulatedElements>
  <details>
    <messages lvl="WARNING" message="Empty element &lt;JournalFileProxyList&gt; was found on line 28"/>
    <messages lvl="WARNING" message="Empty element &lt;ResourceRelationList&gt; was found on line 29"/>
    <messages lvl="WARNING" message="Empty element &lt;ID&gt; was found on line 37"/>
    <messages lvl="WARNING" message="Empty element &lt;Description&gt; was found on line 47"/>
    <messages lvl="WARNING" message="Empty element &lt;Description&gt; was found on line 64"/>
    <messages lvl="WARNING" message="Empty element &lt;Description&gt; was found on line 75"/>
    <messages lvl="WARNING" message="Empty element &lt;DistributionMedium&gt; was found on line 85"/>
    <messages lvl="WARNING" message="Empty element &lt;Price&gt; was found on line 97"/>
    <messages lvl="WARNING" message="Empty element &lt;CollectionType&gt; was found on line 101"/>
    <messages lvl="WARNING" message="Empty element &lt;Multilinguality&gt; was found on line 106"/>
    <messages lvl="WARNING" message="Empty element &lt;Description&gt; was found on line 112"/>
    <messages lvl="WARNING" message="Empty element &lt;Modality&gt; was found on line 127"/>
    <messages lvl="WARNING" message="Empty element &lt;MimeType&gt; was found on line 135"/>
    <messages lvl="WARNING" message="Empty element &lt;MimeType&gt; was found on line 138"/>
  </details>
</xml-validation-section>
<url-validation-section>
  <numOfLinks>6</numOfLinks>
  <numOfUniqueLinks>6</numOfUniqueLinks>
  <numOfResProxiesLinks>1</numOfResProxiesLinks>
  <numOfBrokenLinks>0</numOfBrokenLinks>
  <percOfValidLinks>1.0</percOfValidLinks>
  <details>
    <messages lvl="WARNING" message="URL: https://portal.clarin.inl.nl&#x9; STATUS:301"/>
  </details>
</url-validation-section>
<facet-section>
  <numOfFacets>26</numOfFacets>
  <profile>
    <numOfCoveredFacets>23</numOfCoveredFacets>
    <coverage>0.8846153846153846</coverage>
    <not-covered>
      <facet>lifeCycleStatus</facet>
      <facet>distributionType</facet>
      <facet>rightsHolder</facet>
    </not-covered>
  </profile>
  <instance>
    <numOfCoveredFacets>18</numOfCoveredFacets>
    <coverage>0.6923076923076923</coverage>
    <values>
```

```
<facet name="id">
  <values>
    <value>http://hdl.handle.net/10032/eeba6672493adaa6f24b3bf1c13c385b</value>
  </values>
</facet>
<facet name="_selfLink">
  <values>
    <value>http://hdl.handle.net/10032/eeba6672493adaa6f24b3bf1c13c385b</value>
  </values>
</facet>
<facet name="collection">
  <values>
    <value>INL corpus for contemporary Dutch</value>
  </values>
</facet>
<facet name="projectName">
  <values>
    <value>Reference Corpus of contemporary written Dutch</value>
  </values>
</facet>
<facet name="name">
  <values>
    <value>CHN</value>
  </values>
</facet>
<facet name="continent">
  <values>
    <value>EU</value>
  </values>
</facet>
<facet name="country">
  <values>
    <value>NL</value>
  </values>
</facet>
<facet name="languageCode">
  <values>
    <value>nld</value>
  </values>
</facet>
<facet name="availability">
  <values>
    <value>Free, accessible through CLARIN Institutional login</value>
  </values>
</facet>
<facet name="license">
  <values>
    <value>Free, accessible through CLARIN Institutional login</value>
  </values>
</facet>
<facet name="accessInfo">
  <values>
    <value>Free, accessible through CLARIN Institutional login</value>
  </values>
</facet>
<facet name="organisation">
  <values>
    <value>Institute for Dutch Lexicology (Instituut voor Nederlandse Lexicologie, INL)</value>
  </values>
</facet>
```

```
<facet name="modality">
  <values>
    <value>Written</value>
  </values>
</facet>
<facet name="description">
  <values>
    <value> WORD FORM, LEMMA and PART OF SPEECH</value>
  </values>
</facet>
<facet name="resourceClass">
  <values>
    <value>text</value>
  </values>
</facet>
<facet name="format">
  <values>
    <value>text/plain</value>
    <value>text/xml</value>
  </values>
</facet>
<facet name="nationalProject">
  <values>
    <value>INL corpus for contemporary Dutch</value>
  </values>
</facet>
<facet name="text">
  <values>
    <value>CHN</value>
    <value>corpus hedendaags nederlands</value>
    <value>1.0</value>
    <value>INL</value>
    <value>2014</value>
    <value>1814-01-01</value>
    <value>2014-01-01</value>
    <value>Since 1994, The Instituut voor Nederlandse Lexicologie has put online several corpora of contemporary Dutch: the 5, 27 and 38 million words corpora and the Dutch Parole Internet Corpus. The Corpus Hedendaags Nederlands in the current release is a first step towards a monitor corpus for contemporary Dutch. The material of the old corpora was integrated. For the first release (17 January 2014) a considerable amount of more recent material was added from two newspapers: NRC Handelsblad and De Standaard (until June 2013). For the second release (June 2014) more material from these two sources has been added from July 2013 - December 2013, as has other sources from Suriname and the Netherlands Antilles, such as newspapers, material published on internet (blog, website) and books written by Surinam authors.</value>
    <value>NL</value>
    <value>EU</value>
    <value>Reference Corpus of contemporary written Dutch</value>
    <value>The Corpus Hedendaags Nederlands in the current release is a first step towards a monitor corpus for contemporary Dutch</value>
    <value>Katrien Depuydt</value>
    <value>servicedesk@inl.nl</value>
    <value>http://www.inl.nl</value>
    <value>Documentation in English https://portal.clarin.inl.nl/search/page/help </value>
    <value>English</value>
    <value>eng</value>
    <value>Free, accessible through CLARIN Institutional login</value>
    <value>https://portal.clarin.inl.nl</value>
    <value>Dr. J. Th. Bakker</value>
    <value>Witte Singel/Doelencomplex, Matthias de Vrieshof 2-3, 2311 BZ Leiden, The Netherlands</value>
    <value>jantheo.bakker@inl.nl</value>
    <value>Institute for Dutch Lexicology (Instituut voor Nederlandse Lexicologie, INL)</value>
    <value>31 (0)715272276</value>
```

```
                <value>http://www.inl.nl/</value>
                <value>0</value>
                <value>text</value>
                <value>Monolingual</value>
                <value>Lemma</value>
                <value> WORD FORM, LEMMA and PART OF SPEECH</value>
                <value>Dutch</value>
                <value>nld</value>
                <value>Written</value>
                <value>miscellaneous full texts ranging from legal texts, papers, journals and advertizing. for details see
https://portal.clarin.inl.nl/search/page/help#about</value>
                <value>UTF-8</value>
                <value>text/plain</value>
                <value>text/xml</value>
              </values>
            </facet>
          </values>
          <missingValues>
            <missingValues name="temporalCoverage"/>
            <missingValues name="genre"/>
            <missingValues name="subject"/>
            <missingValues name="_componentProfile"/>
            <missingValues name="keywords"/>
          </missingValues>
        </instance>
    </facet-section>
</instance-report>
```


11.3 Example Collection Report


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<collection-report>
    <timeStamp>2016.02.25.13.50.16</timeStamp>
    <score>59909,870/81543,000</score>
    <avgScore>8,082/11,000</avgScore>
    <file-section>
        <provider>Ethnologue_Languages_of_the_World</provider>
        <numOfFiles>7413</numOfFiles>
        <size>15263840</size>
        <avgSize>2059</avgSize>
        <minFileSize>2045</minFileSize>
        <maxFileSize>2096</maxFileSize>
    </file-section>
    <header-section>
        <profiles count="1">
            <profiles name="p_1288172614026" count="7413"/>
        </profiles>
    </header-section>
    <resProxy-section>
        <totNumOfResProxies>7413</totNumOfResProxies>
        <avgNumOfResProxies>1.0</avgNumOfResProxies>
        <totNumOfResWithMime>0</totNumOfResWithMime>
        <avgNumOfResWithMime>0.0</avgNumOfResWithMime>
        <totNumOfLandingPages>0</totNumOfLandingPages>
        <avgNumOfLandingPages>0.0</avgNumOfLandingPages>
        <totNumOfLandingPagesWithoutLink>0</totNumOfLandingPagesWithoutLink>
        <avgNumOfLandingPagesWithoutLink>0.0</avgNumOfLandingPagesWithoutLink>
        <totNumOfResources>7413</totNumOfResources>
        <avgNumOfResources>1.0</avgNumOfResources>
        <totNumOfMetadata>0</totNumOfMetadata>
```

```
      <avgNumOfMetadata>0.0</avgNumOfMetadata>
   </resProxy-section>
   <xml-validation-section>
      <totNumOfXMLElements>170499</totNumOfXMLElements>
      <avgNumOfXMLElements>23.0</avgNumOfXMLElements>
      <totNumOfXMLSimpleElements>118608</totNumOfXMLSimpleElements>
      <avgNumOfXMLSimpleElements>16.0</avgNumOfXMLSimpleElements>
      <totNumOfXMLEmptyElement>22239</totNumOfXMLEmptyElement>
      <avgXMLEmptyElement>3.0</avgXMLEmptyElement>
      <avgRateOfPopulatedElements>13.0</avgRateOfPopulatedElements>
   </xml-validation-section>
   <url-validation-section>
      <totNumOfLinks>14826</totNumOfLinks>
      <avgNumOfLinks>2.0</avgNumOfLinks>
      <totNumOfUniqueLinks>7413</totNumOfUniqueLinks>
      <avgNumOfUniqueLinks>1.0</avgNumOfUniqueLinks>
      <totNumOfResProxiesLinks>0</totNumOfResProxiesLinks>
      <avgNumOfResProxiesLinks>0.0</avgNumOfResProxiesLinks>
      <totNumOfBrokenLinks>0</totNumOfBrokenLinks>
      <avgNumOfBrokenLinks>0.0</avgNumOfBrokenLinks>
      <avgNumOfValidLinks>1.0</avgNumOfValidLinks>
   </url-validation-section>
   <facet-section>
      <avgFacetCoverageByInstance>0.2692307692307312</avgFacetCoverageByInstance>
   </facet-section>
</collection-report>
```

11.4 Integrated VLO workflow & dashboard - Vision

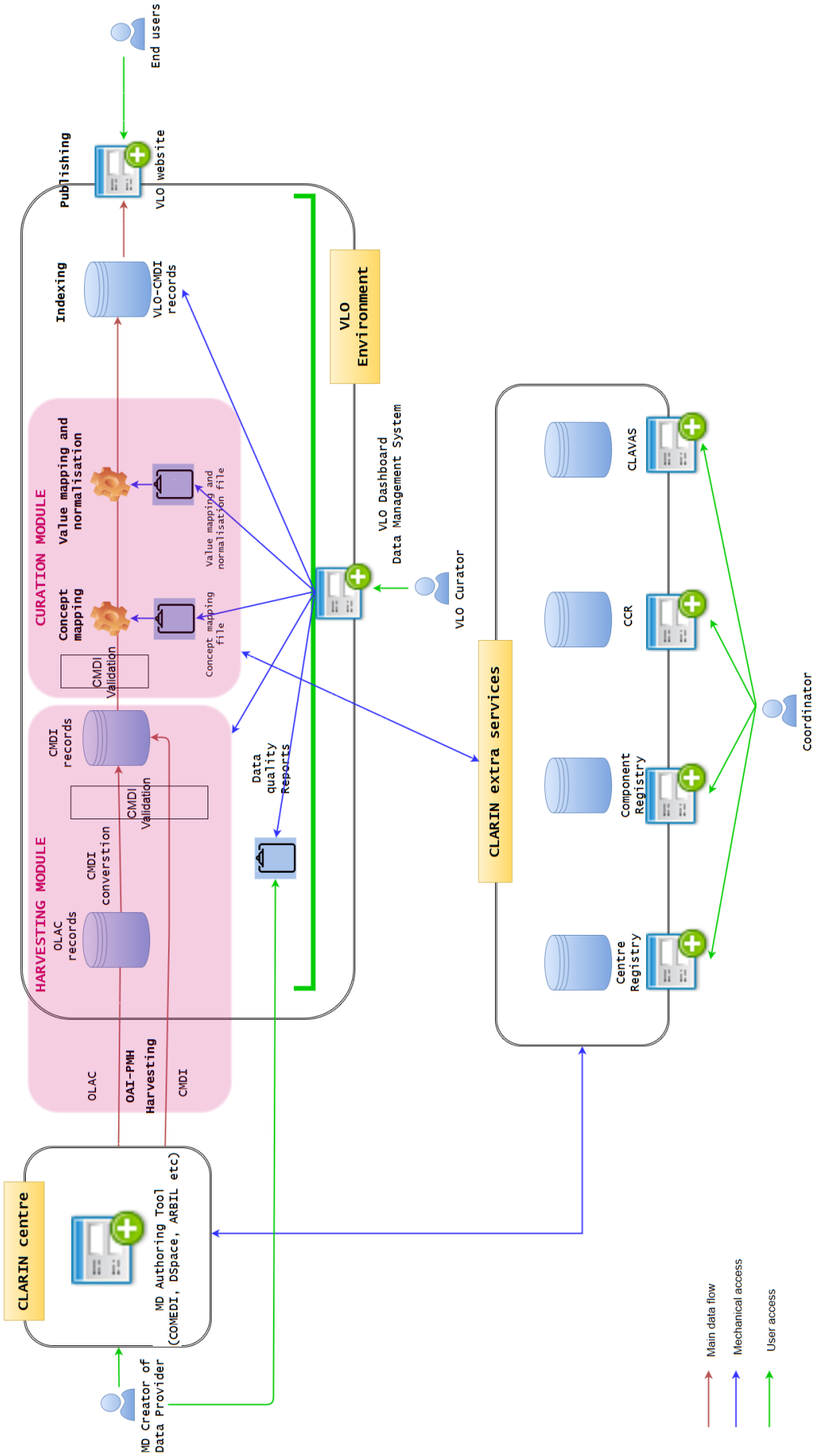

Figure 6: Integrated VLO Workflow with a dashboard for curation and management proposal by ACDH-OEAW (King et al. 2016)

References

1. T. Trippel, D. Broeder, M. Ďurčo, and O. Ohren. 2014. Towards automatic quality assessment of component metadata. In Proceedings of the Ninth International Conference on Language Resources and Evaluation [LREC 2014]. Pp. 3851-3856.
2. M. Kemps-Snijders. 2014. Metadata quality assurance for CLARIN. Technical report.
3. M. King, D. Ostojic, G. Sugimoto, and M. Ďurčo (accepted): Metadata normalisation a case for VLO curation; In Proceedings of the CLARIN Annual Conference 2015,