



**European Holocaust Research Infrastructure
Theme [INFRA-2010-1.1.4]
GA no. 261873**

**Deliverable
D19.5**

Filled Metadata Registry

**Linda Reijnhoudt, Ben Companjen, Mike Priddy
Data Archiving and Networked Services (DANS) - Koninklijke Nederlandse Akademie
van Wetenschappen**

**Tim Veken
Institute for War, Holocaust and Genocide Studies (NIOD)
- Koninklijke Nederlandse Akademie van Wetenschappen**

**Mike Bryant
King's College London**

**Kepa Rodriguez
Göttingen State and University Library**

**Yael Gherman
Yad Vashem**

**Start:
Due: M46
Actual: M54**

Note: The official starting date of EHRI is 1 October 2010. The Grant Agreement was signed on 17 March 2011. This means a delay of 6 months, which will be reflected in the submission dates of the deliverables.

Document Information

Project URL	www.ehri-project.eu
Document URL	
Deliverable	D19.5 Filled Metadata Registry
Work Package	WP19 Data Integration Infrastructure
Lead Beneficiary	P1 NIOD-KNAW
Relevant Milestones	MS4
Nature	P
Type of Activity	RTD
Dissemination level	PU
Contact Person	Mike Priddy mike.priddy@dans.knaw.nl
Abstract (for dissemination)	<p>In this task WP19 imported archival description metadata from 24 sources, which included exports from archival catalogue software, indexing services, structured PDF books, and spread sheets that were provided in 17,744 files representing the holdings of 263 collection holding institutions.</p> <p>The results of this task can be seen in the EHRI portal that is available at https://portal.ehri-project.eu. Software code, documentation and licensing can be found on GitHub at https://github.com/EHRI, with API documentation at http://ehri.github.io/docs/api/ehri-rest/</p>
Management Summary (required if the deliverable exceeds more than 25 pages)	n.a.

Table of Content

INTRODUCTION.....	4
OBJECTIVES OF WP19	4
DATA INTEGRATION STRATEGY.....	5
CENTRAL METADATA REGISTRY	6
METADATA CONNECTORS	6
UNIFORM IDENTIFICATION	6
IMPLEMENTATION	6
PRE-PROCESSING	7
CONVERSION OF MIXED CONTENT TO MARKDOWN EQUIVALENTS	8
PROOF OF CONCEPT FOR AUTOMATION OF THE IMPORT PIPELINE	8
INGEST	9
PRE-PROCESSING	10
IMPLEMENTATION	10
CHAINED TOOLS	11
FILTER AND UNWRAP	11
CONVERSION OF MIXED CONTENT TO MARKDOWN EQUIVALENTS	11
STORING PRE-PROCESSED FILES IN A REPOSITORY	11
IMPORT.....	11
DISCUSSION ON THE PROOF OF CONCEPT	12
SINGLE SOURCE OF RECORDS.....	12
IDENTIFICATION OF DOCUMENTARY UNITS	12
SINGLE INGEST, MANY IMPORTS	12
STEPS AFTER IMPORT	12
PROOF OF CONCEPT CONCLUSIONS.....	13
OVERALL CONCLUSIONS AND RECOMMENDATIONS.....	13

Introduction

The EHRI portal provides access to archival descriptions relating to the Holocaust from archives, libraries and museums from Europe and beyond. These descriptions are either created and edited manually in the portal or provided in digital form and imported into the database that powers the portal.

Collection holding institutes (CHIs) who provide their descriptions to EHRI in a digital form are requested to do so in the Encoded Archival Description¹ (EAD) format following EHRI's Guidelines for Description², which are based on ISAD(G)³. The preferred method of delivery of these files is OAI-PMH⁴, a protocol regularly used for automated harvesting of metadata invoking the HTTP protocol. OAI-PMH supports full and incremental collection of metadata records, notifications of deleted records and downloading specified sets of records. If implemented correctly, EHRI could access the relevant EAD files from the CHIs and update the records in the database. However, this is where theory and practice diverge.

Only a very few CHIs were able to offer records via OAI-PMH and none do it in such a way that makes the import straightforward. Differences between the above ideal situation and reality include:

- There is no set that contains just the records that had been selected for EHRI. This means the records have to be filtered after the harvest, or individual records need to be retrieved.
- The OAI-PMH endpoint behaves in unexpected ways. One endpoint did not stop sending records after the set of records had been downloaded completely, but kept sending "there is more" and started to repeat records.
- The XML⁵ does not use XML Namespaces correctly, so that the actual metadata was not separable from the container XML specified by OAI-PMH.
- The EAD has content issues. In one example, descriptions of parts of a documentary unit were in separate records instead of being nested in an EAD hierarchy. In other cases, the EHRI (or ISAD(G)) guidelines were not followed and EHRI had to reorder metadata in the records.
- The database loader API that imports and inserts or updates the metadata from EAD files into the graph database⁶ cannot handle some valid EAD elements. The current implementation of the EAD importer cannot import mixed content correctly, so text paragraphs with EAD mark-up elements do not import as expected.

Objectives of WP19

The objectives of this work package relevant to this deliverable are:

¹ <http://www.loc.gov/ead/>

² See deliverable D17.3 – Guidelines for the application of standards, second version.

³ General International Standard Archival Description: <http://www.ica.org/10207/standards/isadg-general-international-standard-archival-description-second-edition.html>

⁴ Open Archives Initiative Protocol for Metadata Harvesting: <http://www.openarchives.org/pmh/>

⁵ Extensible Markup Language: <http://www.w3.org/XML/>

⁶ See deliverable D19.2

- To define a data integration strategy that enables central discovery and use of content from distributed heterogeneous collections for use by the VRE⁷ (Work Package 20).
- To provide metadata connectors which collect and translate the metadata from the collections to the standardized metadata in the registry.

The filled metadata registry is the culmination of the work carried out by WP19, but has also become considerably more work than originally expected. All the metadata provided in a digital format be it as PDF, spread sheet, csv, or in some XML schema, was unique to the provider, with no common formats or structure. Therefore every connector, which collects, translates and harmonises, is bespoke to the metadata provider's specific context and provision format.

WP19 imported archival description metadata from 24 sources, which included exports from archival catalogue software, indexing services, structured PDF books, and spread sheets that were provided in 17,744 files representing the holdings of 263 CHIs.

The results of this task can be seen in the EHRI portal that is available at <https://portal.ehri-project.eu>. Software code, documentation and licensing can be found on GitHub at <https://github.com/EHRI>, with documentation at <http://ehri.github.io/docs/api/ehri-rest/> and the tools to harmonise and standardise the heterogeneous metadata that was being supplied can be found here <https://github.com/EHRI/ehri-ead-preprocessing>.

In addition a proof of concept for automated importing of archival descriptions was demonstrated for a single CHI.

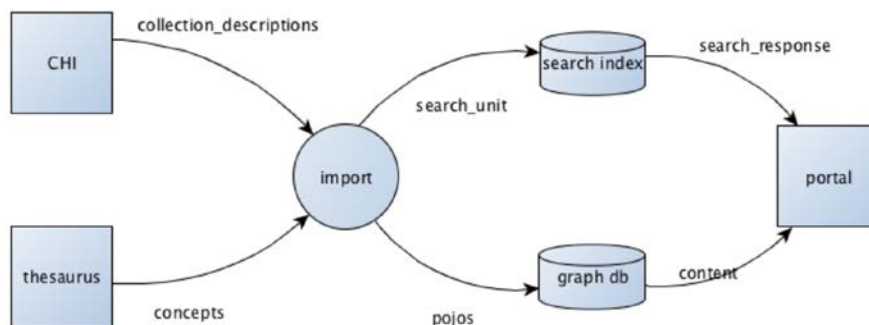
Data integration strategy

This section summarises the deliverable D19.2 (Metadata Registry), which describes the metadata registry and the proposed processes for importing into the graph database that sits behind the portal site.

A harvesting strategy has been chosen to aggregate metadata from CHIs⁸. The metadata formats have been mapped towards an internal schema. Named entities have been connected via a lookup strategy.

⁷ Virtual Research Environment

⁸ See deliverable D19.1 State of the Art Report for background to this choice.

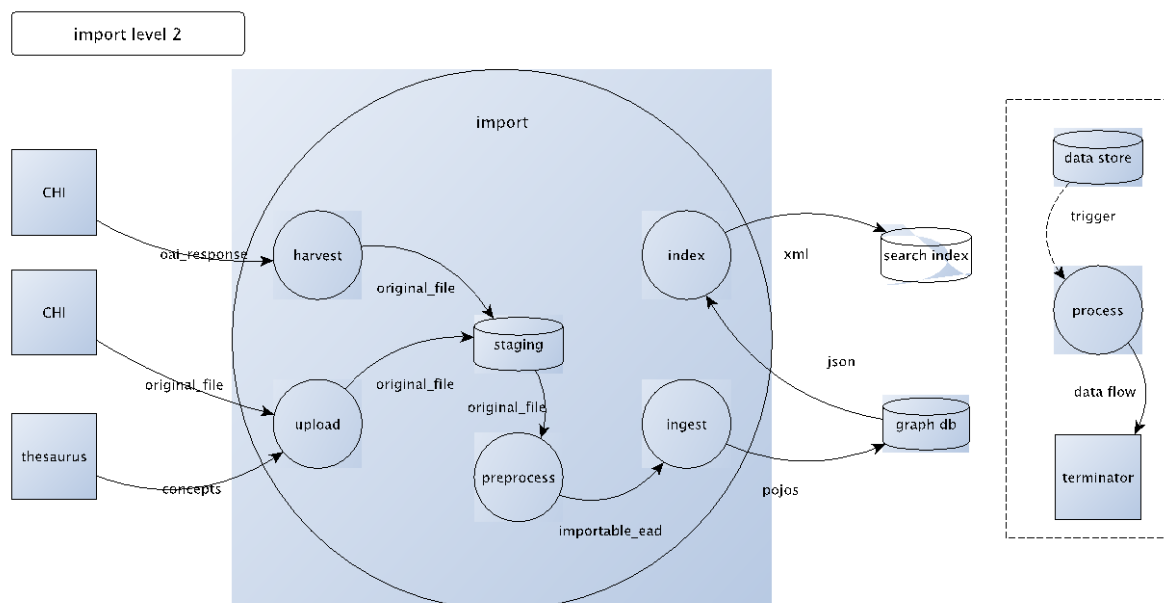


Central metadata registry

The mapped metadata has been imported into a central database and connected to previously imported metadata.

Metadata connectors

Have been developed to aggregate, map, connect and store the metadata in the central metadata registry.



Uniform identification

Wherever possible the identifiers available and provided by the CHI have been used to create unique identifiers within the central database.

Implementation

Metadata has been aggregated from CHIs using OAI-PMH when available (see Proof of Concept below). When no standardised strategy for data sharing was available, metadata was transferred in methods that are one-time-only.

The CHIs use different formats to describe and export their metadata, including EAD, DC and various non-standardised formats. The mapping transformed these into the internal schema that is based on EAD.

In order to execute the mapping pre-processing steps were required. The basic ones focus on well-formedness and valid XML. Critical to the system are the unique identifiers. When CHIs did not provide those on all levels of descriptions, newly created identifiers were added. These pre-process workflows contained both generic and CHI-specific steps. Connections have been created during import between the new entities and previously imported entities.

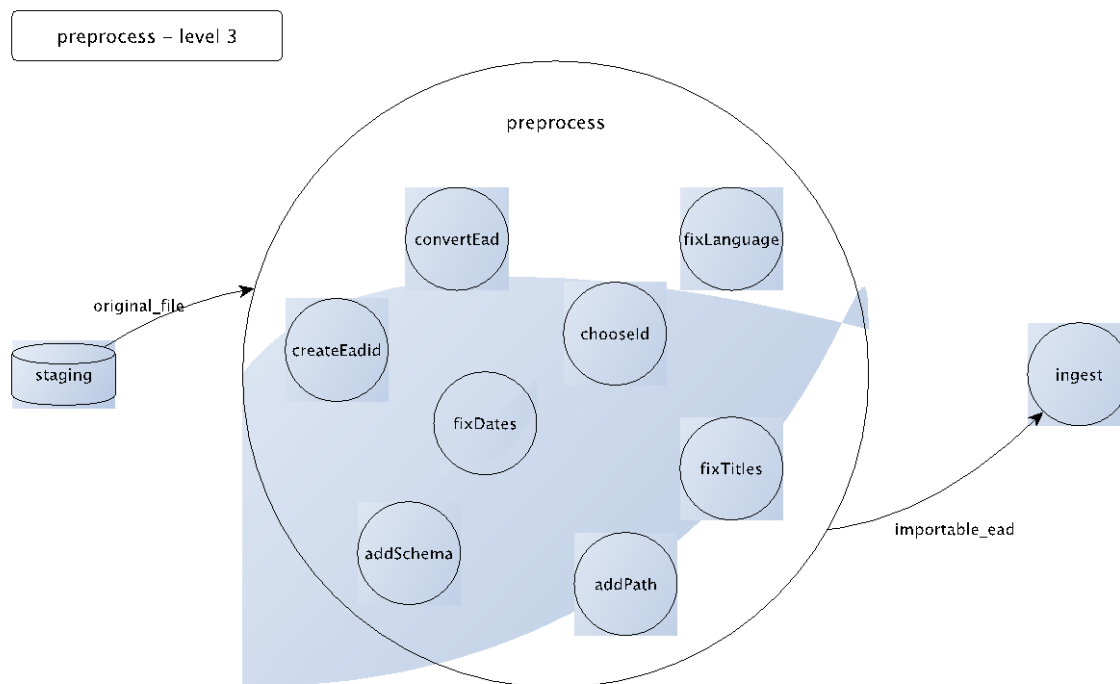
Due to the tailor-made connections and many manual adjustments, the current system is not sustainable and will desynchronise with the descriptions held at the metadata providing CHIs.

Pre-processing

EHRI has produced various pre-processing tools to normalise or even correct records prior to ingest, for example correcting date format and language code. These tools are used prior to importing the data into the database using the importer tools, because the importer tools have requirements that not all records adhere to.

Metadata from specific CHIs need more pre-processing than from others. Conversion pre-processing tools will undertake bespoke transformations converting data formats provided to EAD. The order in which this suite of tools is used is essential to ensure correct transformation and normalisation. Some tools are XSLT stylesheets, whilst others are Java processors packaged as a JAR file.

Rather than creating a new tool with functionality unique to a metadata provider, the process has been broken down into elemental parts and where possible generalised. Therefore a selection of more generic tools and some bespoke tools can be ordered into a workflow, that integrates the existing tools required and configures routes from ingest to import via the tools. In the Proof of Concept below tools have been chained to automate the import process.



Conversion of mixed content to Markdown equivalents

The EAD importer in the backend of the portal has a few limitations. The translator integration pattern is implemented here to make sure the incoming files are formatted in a way that the importer can use, while still being valid EAD. The limitation we are circumventing in this step is the lack of support for mixed content, such as:

<p>There is <emph render="italic">italics</emph> in my text. </p>

Three style sheets translate this and similar mark-up to Markdown⁹ mark-up, which is how the portal interprets the text. Not all source records include the entire possibly problematic mark-up, and in future iterations the importer may correctly parse the mixed content. Depending on the source, these translator steps can be included or excluded from the pre-processing workflow.

Proof of Concept for Automation of the import pipeline

As already indicated, no CHIs were able to publish standards compliant metadata about their archival holdings and only few have an OAI-PMH services from which to harvest, therefore we were only able to demonstrate a proof-of-concept of a possible method by which the portal is synchronised with the archival descriptions of the CHIs.

This section outlines the setup and use of open source components and the WP19 developed pre-processing tools, chained to automate the ingest, pre-processing and import of EAD files from a single collection holding institution. We will outline the component that manages the ingest process from OAI-PMH endpoints and folder, the

⁹ <http://daringfireball.net/projects/markdown/>

integration of various pre-processing components into a tool chain and the automated import of metadata records. Finally the last section will discuss some outstanding issues and limitations.

Ingest

The proof of concept (PoC) starts with the ingest process. This is handled by the REPOX¹⁰ software, which can ingest XML metadata from OAI-PMH endpoints, folders on a file system, HTTP¹¹ endpoints and FTP¹² servers. It can also act as an SRU¹³ Update endpoint to get updated records pushed to it. REPOX is currently maintained by Europeana¹⁴, who use it in the infrastructure of The European Library to harvest bibliographical metadata from national and research libraries around Europe. REPOX has a web interface and a REST¹⁵ interface to manage the (minimal) administrative information about data providers and mostly technical information about the data sets they provide. REPOX can ingest records on demand, or following a schedule. Similarly, exporting the records to a folder on the file system can happen on demand or following a schedule.

¹⁰ <http://labs.europeana.eu/apps/REPOX/>

¹¹ Hypertext Transfer Protocol: <http://www.w3.org/Protocols/>

¹² File Transfer Protocol: <https://www.ietf.org/rfc/rfc959.txt>

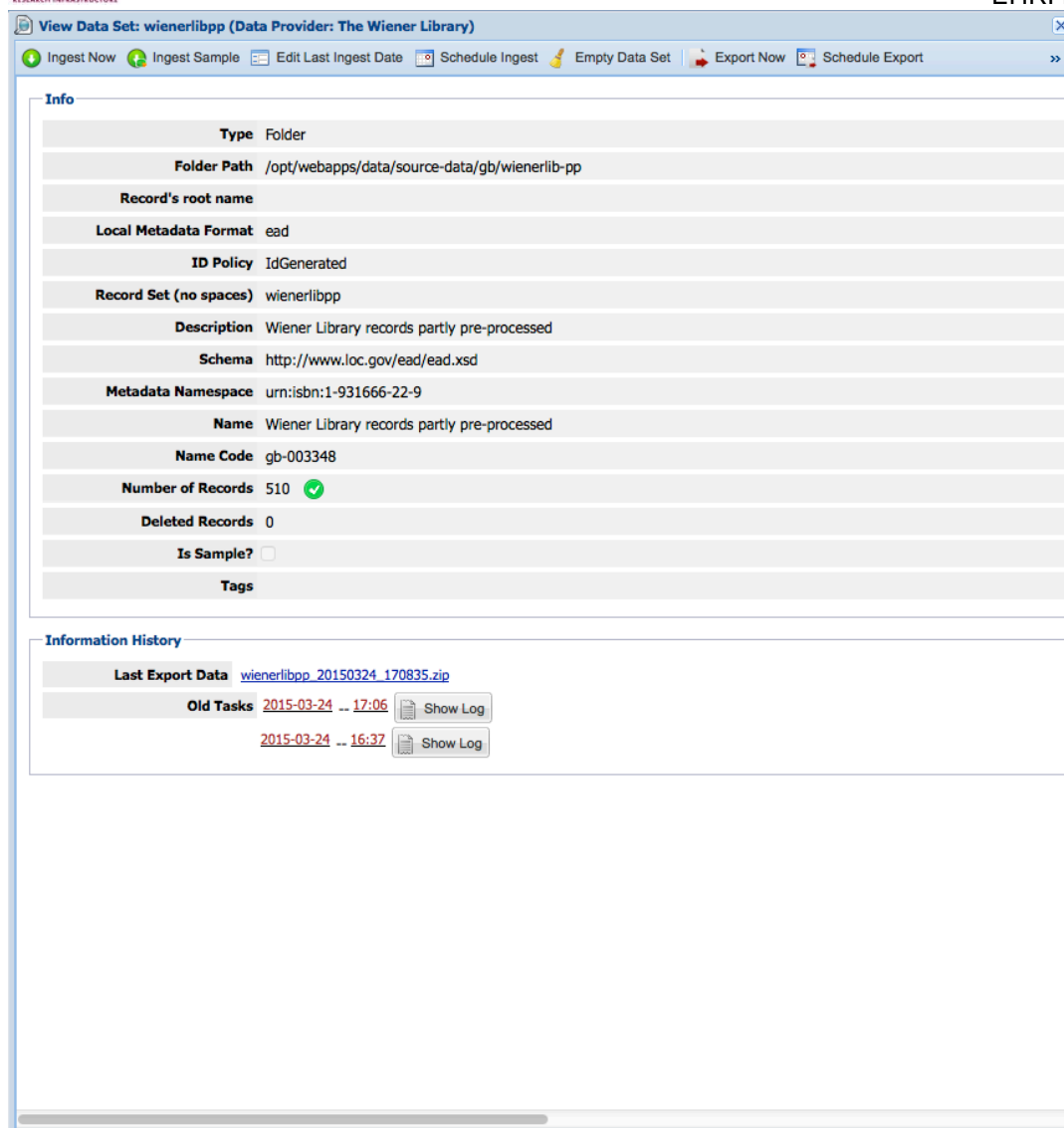
¹³ Search/Retrieval via URL: <http://www.loc.gov/standards/sru/index.html>

¹⁴ <http://www.europeana.eu/portal/>

¹⁵ Representational State Transfer:

https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm,

http://en.wikipedia.org/wiki/Representational_state_transfer



View Data Set: wienerlibpp (Data Provider: The Wiener Library)

Ingest Now | Ingest Sample | Edit Last Ingest Date | Schedule Ingest | Empty Data Set | Export Now | Schedule Export

Info

Type	Folder
Folder Path	/opt/webapps/data/source-data/gb/wienerlib-pp
Record's root name	
Local Metadata Format	ead
ID Policy	IdGenerated
Record Set (no spaces)	wienerlibpp
Description	Wiener Library records partly pre-processed
Schema	http://www.loc.gov/ead/ead.xsd
Metadata Namespace	urn:isbn:1-931666-22-9
Name	Wiener Library records partly pre-processed
Name Code	gb-003348
Number of Records	510 ✔
Deleted Records	0
Is Sample?	<input type="checkbox"/>
Tags	

Information History

Last Export Data	wienerlibpp_20150324_170835.zip
Old Tasks	2015-03-24 -- 17:06 Show Log 2015-03-24 -- 16:37 Show Log

Pre-processing

This tool chain can be created by integration of various pre-processing tools. Rather than creating a new tool with the functionality of the tools needed for each CHI's records, integrating the existing tools requires configuring routes from ingest to import via the tools.

Implementation

Software programs and tools can be integrated by viewing their inputs and outputs as messages that are passed around. Common scenarios have been identified as Enterprise Integration Patterns. Many of these patterns, if not all, are supported by the Apache Camel framework¹⁶. We chose to use the Apache ServiceMix application¹⁷ (version 5.4.0) to dynamically deploy a proof of concept route from the REPOX ingest manager to the REST service for importing pre-processed EAD files.

¹⁶ <http://camel.apache.org>

¹⁷ <http://servicemix.apache.org>

Apart from the basic installation of ServiceMix, the implementation depends on OSGi¹⁸ features camel-saxon¹⁹ and camel-http²⁰, which can be installed from within ServiceMix.

Chained tools

The tool chain is file based, meaning the inputs and outputs of each step are files in the file system. Other exchange mechanisms are supported, such as message queues and shared databases, but for the purpose of this demonstration we tried to minimise the number of dependencies.

Filter and unwrap

As REPOX manages records from various sources, including OAI-PMH, it outputs ("exports") all records in the set, including records that in OAI-PMH were marked as 'deleted'. All exported files are wrapped in a REPOX wrapper, but the deleted records should not be included in the processing steps. The files that do contain EAD must have their REPOX wrapper removed, which is the filter step in the Camel route.

Conversion of mixed content to Markdown equivalents

In the proof of concept, three XSLT style sheets are used to convert `<emph>` within `<p>` to Markdown, convert `<extref>` to Markdown links and remove `<emph>` in `<persname>` to a concatenation of name parts.

Storing pre-processed files in a repository

The final step before import is management of the processed files. In this PoC processed files are renamed and stored in directories based on the country and repository codes in the EAD files.

If the tool chain is used again with updated files, the EAD structure in the new processed files could be compared to that in the existing files to see if structural changes occurred. The import process, especially if some units get different identifiers, does not support changes in the structure of archival fonds. Content updates of existing documentary units are supported.

Import

The EHRI portal backend allows calling an HTTP endpoint to start an import process. A call to this endpoint requires a number of parameters: the scope of the imported unit description(s), i.e. the identifier of the repository; a log message about the import, which the import guidelines require to include the source and date of ingest; and a properties file that specifies the mapping of EAD fields to description fields defined by the EHRI guidelines.

The PoC defines hardcoded constant values for each of these parameters, but it is possible to use a registry of source datasets and accompanying values to dynamically find the correct values for each input.

In this PoC, the end of the tool chain is this endpoint that accepts a single EAD file as the

¹⁸ <http://www.osgi.org/Main/HomePage>

¹⁹ <https://github.com/apache/camel/tree/master/components/camel-saxon>

²⁰ <http://camel.apache.org/http.html>

body of a POST request. As each import creates an import event in the database, using this endpoint creates some overhead. To prevent overloading the server, the import is throttled at a maximum of two import calls per second.

Discussion on the Proof of Concept

This PoC demonstrates the use of Enterprise Integration Patterns²¹ for chaining various pre-processing tools in a tool chain from ingest to import. As it is a PoC, there are some limitations, however, we believe these can be overcome in future development iterations.

Single source of records

Although EHRI has many collection holding institutes who provide metadata records, only a few were able to provide a consistent level of metadata quality over time. Efforts were often put into helping the CHIs provide better metadata, which means that new or different pre-processing was needed after updates. The variety of delivery methods, combinations of tools, and regular manual effort, made defining tool chains labour intensive. REPOX could have been used to manage datasets and ingests, but the integration tool chain was not yet ready and using REPOX without the integration requires only more effort rather than less.

This proof of concept was created around the records from a single CHI who need some fairly easy pre-processing to overcome a limitation of the importer, but it also shows that automated import is possible.

Identification of documentary units

In the current implementation of the tool chain, some essential steps are missing. Not mentioned above is the requirement that every documentary unit needs to have an identifier that is unique within its parent scope. As some EAD files do not have <unitid> elements whose values are unique within the scope for every unit whilst other EAD files have multiple unit identifiers, two processes were created to make sure every unit has a <unitid> and a single <unitid> is selected as the main identifier. The tools used for these processes did not integrate easily into the tool chain, so the ingested records were manually pre-processed with these tools.

Single ingest, many imports

REPOX's process model links a single ingest event (e.g. a harvest or import-from-folder action) to the set of files in that ingest. Similarly, it exports the files in a dataset in a single export event. Processing and importing each file individually decouples the single export event from the import events that follow. This also makes tracing provenance difficult. Further development could include adapting the import part of the tool chain to use the import endpoint that takes a list of file names and imports all listed files in one event. The accompanying log message could also be created dynamically from the information about the last ingest action in REPOX to connect ingest and import.

Steps after import

After each update in the database, the index that is used to represent the item in the front-end also requires updating. After a manual import, a tool that performs the re-

²¹ <http://www.eaipatterns.com/index.html>

indexing is called. This PoC, however, does not perform the necessary re-indexing of records after an import, because the decoupling of ingest and import makes it harder to determine the right moment for indexing. The importer does not provide enough information about the result of an import to determine this moment. If in the future all files are imported in a single call, it should become easier to re-index automatically after an import.

Proof of Concept Conclusions

We described a proof of concept implementation of a chain of tools that was successfully used to automatically ingest, pre-process and import EAD files from a collection holding institute. The current implementation can serve as the basis for more automation in the ingest process, pre-processing and import of archival metadata from a larger number of CHIs.

Overall Conclusions and Recommendations

Through the work on this task WP19 has built considerable knowledge about how CHIs address the process of describing their archival holding and the possibilities of sharing or publishing this information. It is clear that providing machine-readable metadata to an external research infrastructure is novel and challenging. Without doubt we were impressed with the number of institutions willing to work to provide this information, however WP19 were often putting in effort to aid the CHIs in creating better, more standards compliant, metadata. Undertaking this necessary preparatory remedial work, has taken time away from ensuring sustainability through developing automated process for producing well-formed EAD and publishing it.

Handling many formats of metadata delivered and the development of a set of pre-processing tools that could be chained provides a platform for future sustainable iterations of the EHRI portal where the CHIs are in charge of their own metadata publishing. This would require CHIs to implement standardised protocols for sharing metadata, using internationally standardised metadata exchange formats correctly and by using unique persistent identifiers on all levels of description. Ultimately the metadata registry & import process should only accept metadata that conforms to international standards, and thus meeting the fourth aim of this work package namely: to stimulate and facilitate uniform identification of, and access to, the collections.

To facilitate the sharing of metadata CHIs must:

- Improve the quality of the metadata it holds including ensuring unique identifiers are present at all levels of description,
- Document the metadata lifecycle,
- Ensure policies and clearly defined & documented procedures, for creating and publishing archival descriptions to the defined standards, are in place,
- Ensure that policies and procedures are communicated to all relevant staff,
- Define roles and responsibilities within their organisation for policy and quality,
- Include the above into existing documenting workflows.

This may require a lot of effort, and change management, for many collection-holding institutions, but it is necessary if they wish to remain current and relevant in this digital joined-up world.

