ECP 2008 DILI 518002 EUscreen

Exploring Europe's Television Heritage in Changing Contexts

D4.4 - Report on EUscreen Web Services

Deliverable number	*D4.4 - Report on EUscreen web services*
Dissemination level	*Public*
Delivery date	*15 February 2011*
Status	*Final*
Author(s)	*Vassilis Tzouvaras,Kostas Pardalis, NTUA*

e*Content*plus

This project is funded under the *e*Content*plus* programme[1]
a multiannual Community programme to make digital content in Europe more accessible, usable and exploitable.

1 OJ L 79, 24.3.2005, p. 1.

Document Information

Deliverable number:	*D4.4*
Deliverable title:	*Report on EUscreen web services*
Actual date of deliverable:	M14
Workpackage:	4
Workpackage title:	Semantic Access & Integration
Workpackage leader:	NTUA
Keywords:	Ingestion tool, metadata mappings, statistics
Internal reviewer:	Johan Oomen

Table of Contents

DOCUMENT INFORMATION **2**

TABLE OF CONTENTS **3**

1 SCOPE OF THIS DOCUMENT **4**

2 INTRODUCTION **5**

3 INGESTION ANALYSIS AND WORKFLOW **7**

4 HARVESTING AND DELIVERY SERVICE **9**

4.1 IMPLEMENTATION DETAILS 9
4.2 CONTENT UPLOAD 9
4.3 USER AND ORGANISATION MANAGEMENT 9
4.4 FUNCTIONALITY AND USER INTERFACES 10

5 MAPPING MODULE **17**

5.1 FUNCTIONAL ANALYSIS 17
5.2 MAPPINGS 20
5.2.1 Source schema *20*
5.2.2 Target Schema & Mapping Area *21*
5.2.3 Mapping types *22*
5.2.4 Conditions *23*
5.2.5 Preview & Mapping Summary *23*
5.3 SOFTWARE INTERFACES 25

6 STATISTICS **28**

6.1 FUNCTIONAL ANALYSIS 29
6.2 SOFTWARE INTERFACES 33

7 TRANSFORMATION SERVICES **36**

7.1 TRANSFORM 36
7.2 REVIEW TRANSFORMED DATASET 37

8 ANNOTATION SERVICES **38**

9 RELEVANT WORK **43**

9.1 THE HP-MIT DSPACE REPOSITORY PROJECT 43
9.2 THE FEDORA DIGITAL OBJECT REPOSITORY MANAGEMENT SYSTEM 44
9.3 THE EPRINTS REPOSITORY PLATFORM 45
9.4 THE CERN DOCUMENT SERVER SOFTWARE (CDSWARE) 45
9.5 DRIVER: BUILDING A SUSTAINABLE INFRASTRUCTURE OF (EUROPEAN) SCIENTIFIC REPOSITORIES . 46
9.6 REPOX – A METADATA SPACE MANAGER 47

10 CONCLUSION **48**

1 Scope of this document

This document constitutes the first report for the design and development of the web services that will be used for the ingestion and publishing of cultural content metadata belonging to providers that participate in the EUscreen project. This process aims to guarantee semantic interoperability across numerous data repositories of varied thematic categories, technical features and capabilities, allowing seamless ingestion of diverse content and knowledge. The results of requirements analysis together with the outcome of close cooperation with relevant work packages and working groups of the project are illustrated and, the subsequent decision making process that has lead to the final functional and technical requirements of the prototype are discussed. Additionally, the core modules that currently serve the ingestion process are described, together with examples and screens that cover all the functionality of the implemented web services. This document can be used as a reference for the prototype design and implementation as well as for providing instructions for the service's usage, modules and features.

The web-services is developed to support the consortium needs through close cooperation with all relevant work packages. Primary focus is on supporting aggregation of arbitrary provider organisation data models, adopting the EUscreen Metadata Schema as the reference metadata schema and, storing and publishing content using the EBUcore Metadata standard in Europeana and other external web applications. The basic steps in the process that leads to semantic interoperability and allows for the ingestion and aggregation of all cultural heritage content within EUscreen and the subsequent publishing of the semantically interoperable metadata, specifically for harvesting by the Europeana portal, are as follows:

- registration and access rights for users and their respective organisations, supporting national or thematic aggregators
- import and parsing of organisations' metadata records, supporting any proprietary or standardised schema
- analysis (statistical, structural and semantic) of user input in order to provide a detailed overview and to assist the user in subsequent steps with previewing and guiding capabilities
- cross-walk editing and transformation of user metadata records to a reference, well defined schema that will allow for bidirectional interoperability with all standardised outside sources; focus of deployment is interoperability with Europeana

Edit or create new user metadata records as part of the ingested metadata records set, using the annotation services that are integrated in the ingestion tool.

2 Introduction

WP4 is responsible for the semantic alignment of metadata schemata that content providers use to annotate their items, to a common, well-defined, machine understandable schema, which in turn will allow for the ingestion of aggregated content in the EUscreen and the Europeana portals. Content providers currently participating in the EUscreen project, as well as several that are contemplating their contribution during the project's life cycle, constitute mainly of audiovisual archives. There are a limited number of key standards, and they are used extensively throughout Europe and indeed the world. These are often suggested as best practice but, there is still a long way to go to achieve interoperability. National standards in some countries are also a factor that needs to be taken into consideration. There is also often the case where providers bypass the semantics of a standard through proprietary rules and conventions that are not sufficiently documented or semantically defined, resulting in misinterpretations that can not be straightforwardly detected. Finally, there is a vast variety, with respect to the level of detail, in annotation, ranging from the use of a small flat-structured set of elements to defining and using complex schemata that support various levels of annotation, item and concept relations and connections with controlled vocabularies and thesauri.

The adoption of EUscreen Metadata Schema, coupled with the loosely defined providers' input schemata, led WP4 to design an aggregation and mapping workflow that will allow for an elaborate, visually guided ingestion of metadata in the repository. The fundamental principles include the disassociation of input metadata from existing metadata standards in order to avoid ambiguity over interpretation and, the ability to create and manage transformations that will apply to the actual metadata records, which subsequently (re-)define the input schema in a semantic, machine understandable way, based on its mapping to the EUscreen Metadata Schema. Following this process, the metadata are stored in EBUcore format to achieve interoperability with external systems. Finally, a mechanism is facilitating the export of the metadata n EBUcore for Europeana.

The web services aim to provide a user friendly ingestion environment that allows for the extraction and presentation of all relevant and statistical information concerning input metadata together with an intuitive mapping service that illustrates the EUscreen Metadata Schema and its mapping to EBUcore standard and provides all the functionality and documentation required for the providers to define their crosswalks. Transformations are editable and reusable and can be applied incrementally to user input while providing, throughout all steps, best practice examples, previews and visual indications to illustrate and guide user actions. One of the key capabilities lies in the ability to semantically enhance user metadata through conditional mapping of input elements using respective transformation functions (e.g. concatenate) that will allow for the addition and enrichment of semantics even when those are not specifically stated in the input.

The services determine the operational workflow processes to bring the amalgamated content of the partner audiovisual archives into thee EUscreen portal and Europeana. It is also the technical baseline to create, manage and execute, with the European Digital Library Office, the implementation plan to ensure that the content is visible in Europeana. The platform

supports exporting of the aggregated metadata to several established standards concerning presentation and archive management. Primary effort is directed to the transformation of the aggregated content to the Europeana Data Model and the deployment of an OAI-PMH repository to facilitate harvesting by Europeana.

The rest of this document is structured as follows:
Chapter 3 illustrates the Ingestion Analysis and the Workflow that was designed for the EUscreen project, illustrating the processes a audiovisual content provider undertakes to enable the presentation of its content through the EUscreen portal. Chapter 4 describes the Harvesting and Delivery service that is responsible for setting the environment, user roles and access rights to define the tasks that users can perform on specific organisations' repositories. It is the basis of the whole service and gives access to the functionality of the system. Chapter 5 outlines the Mapping module as it is set up to support EUscreen Metadata Schema as an intermediate schema and EBUcore as the sorting and exporting standard, together with relevant functionalities that enable semantic interoperability for the ingested content. Chapter 6 gives an overview of the Statistics module and relevant information and functionality that it presents to the user guiding the mapping process. Chapter 7 describes the transformation services. Chapter 8 describes the annotation service. Chapter 9 contains an overview of relevant platforms and tools that deal with ingesting, mapping and transforming metadata records as well as with enabling permanent access to digital works. Finally, Chapter 10 summarizes the Conclusions of the report.

3 Ingestion analysis and workflow

In the Cultural Content Metadata Space, the largest technological challenge is to ensure syntactic and semantic interoperability across the different types of metadata that exist in the Cultural Heritage sector. The technical standards enabling interoperability form an important dimension of this work. In order to achieve semantic interoperability we need a common automatic interpretation of the meaning of the exchanged information, i.e. the ability to automatically process the information in a machine-understandable manner. The first step of achieving a certain level of common understanding is a representation language that exchanges the formal semantics of the information. Then, systems that understand these semantics can process the information and provide web services like searching, retrieval etc.
The following figure illustrates the proposed workflow for ingesting metadata in EUscreen.

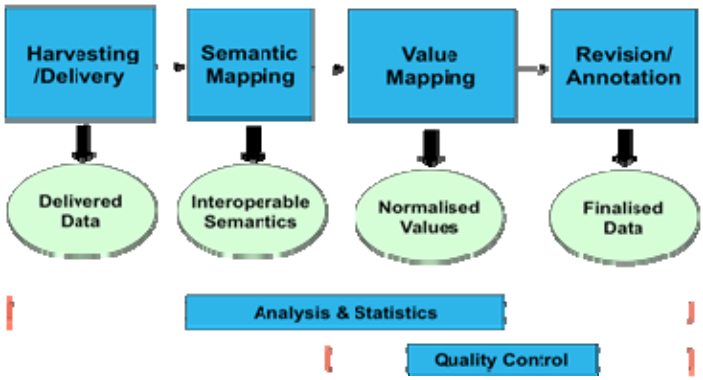


Figure 1 ingestion workflow

The workflow consists of four phases. Each phase is responsible for specific services all needed to ensure the quality of the ingestion process.

[1] Harvesting/delivery is responsible for collecting the metadata It will be an interface for different methods of data delivery including, OAI-PMH, HTTP upload/download FTP upload/download

[2] Semantic Mapping will provide the service for assigning semantics to the harvested metadata. It will assist to manually map Providers fields to a rich reference schema. Providers that have metadata in supported known formats might be able to omit this step (use stored transformations from selected schemas to the reference schema based on existing crosswalks).

[3] Value Mapping will take existing attribute values and produce different/edited values. In particular:

1. It will enable providers to resolve data issues, e.g. map own terminology list to selected terminology lists
2. It will then automatically normalize data e.g. dates, geographical locations, nationality/language, name writing convention to selected vocabulary standards.

[4] Revision/Annotation will enable the addition of data that is not in the original metadata (e.g empty fields, fields that take values from controlled vocabularies).

Two additional modules include:

Analysis & Statistics service will provide detailed analysis and statistics of metadata contributed by a provider. (i.e. number of items imported, total values per field etc).

Quality Control will automatically check and report on Content Provider's data (i.e. missing values, malformed data). Error reports and warnings will be produced to facilitate editing the semantic mappings, value mappings and/or edit items until the Provider's data successfully passes the Quality control checks.

4 Harvesting and delivery service

4.1 Implementation details

The system is implemented as a web service, where authentication is required to perform a series of tasks that correspond to work flow steps. The service is an application written in the Java programming language and hosted on a web server by the Tomcat servlet engine. Data is imported into a PostgreSQL database in xml format (as BLOB).

Once uploaded, the xml structure is parsed and represented in a relational database table. As this table can grow quite large it is partitioned into one partition per data upload. All data within one upload is treated as having the same structure, so it is not possible to upload different schemas (or more likely updated schemas) in one upload.

Most of the communication between the application and the database is implemented on the Hibernate framework, a high performance object/relational persistence and query service. This allows for powerful, yet simplified, management of housekeeping objects like Users, Organizations and Data Uploads while also providing additional functionalities such as integration with Lucene for indexing and querying data.

Once data is parsed into the relational table, indexes are built to allow quick access to any part or sub-tree of the xml-tree like data. These are currently constructed as PostgreSQL BTREE indexes; when content full text indexing is implemented, it will be based on Hibernate's search architecture. All further data manipulation such as mapping and transformation, normalization, enrichment, etc. is structured through the addition of extra tables annotating but not altering the original data. This allows easier comparison between uploads and facilitates the versioning strategy.

4.2 Content upload

Currently, allowed data formats for uploads are:

- XML in any schema.
- zip archives of the above

Uploading of thumbnails or really any binary data will be supported in the future.

Following methods are supported for uploading content:

- HTTP upload; suggested only for relatively small amounts of data (<2MB)
- Upload to a dedicated FTP server.
- Remote HTTP or FTP browsing.
- OAI-PMH repository harvesting.
- SuperUser uploading from local file system (restricted).

4.3 User and organisation management

Users belong exclusively to one organization and can not access data non related to that. Users can be assigned with different levels of access, that grant roles ranging from data

browsing, over editing and annotating, to being allowed to edit other users' details (administrators). We have extended user roles to allow parent users, for organizations that might not have expertise or manpower to use the system and thus, delegate the job to an organization which is then their designated "parent" organization. Parent users extend their rights to child organizations and provide the functionality to build the access hierarchy for any given country/thematic category. The current role set can be easily adjusted to allow more freedom in rights management.

The following rights are currently implemented:

- change/add/delete user
- change/add/delete organization
- edit/upload/delete data
- publish / declare finished datasets
- read-only browsing rights

These rights have been grouped to the following roles:

- Administrators (all user, data and organization rights for the organizations they manage),
- Annotators (data management rights),
- Publishers (publishing data rights),
- Data Viewers (simple viewing rights).
- The system also contains some hardcoded super-users that have full rights to all organizations and their data in the system.

4.4 Functionality and user interfaces

Users can join the EUscreen service using the registration page (fig. 4.1). During registration they are prompted to select the organization they belong to.

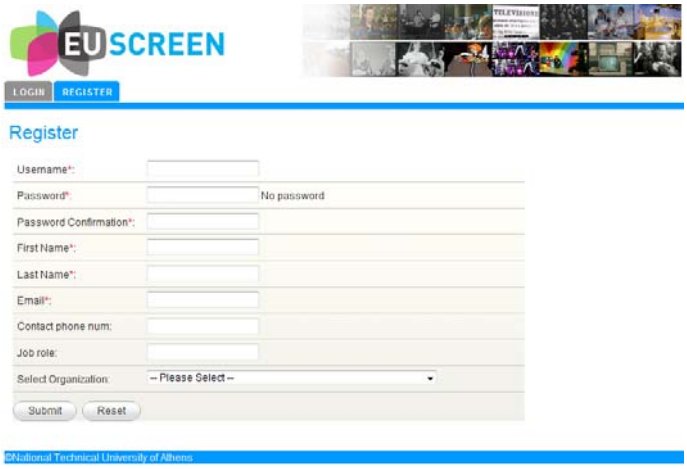


Figure 4.1 Registration Screen

The administrator of that organization is notified by email for the pending user registration and is authorized to grant the appropriate rights and finalize the procedure.

In case a user's organization is not present in the list of registered organizations, the user can register in the syste. In this scenario he is given the opportunity to create a new organization and automatically become the administrator for it.

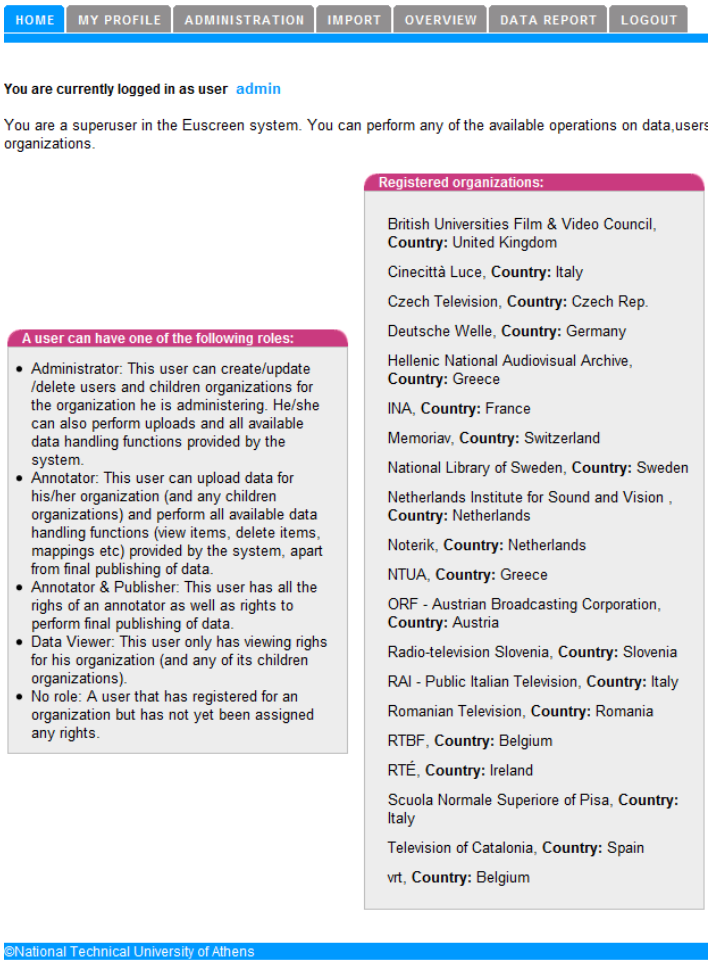


Figure 4.2 Home screen

Organizations within the system can have parental organizations. Users of parental organizations extend their rights to the children of the organization (and in turn to grandchildren that may exist, and so on). Every organization can have at most one parent organization. The parent organization has to agree on publishing the data of the child organizations (among other things). This way an aggregator for example can define and manage all the organizations (and their respective data) he is supervising within EUscreen.

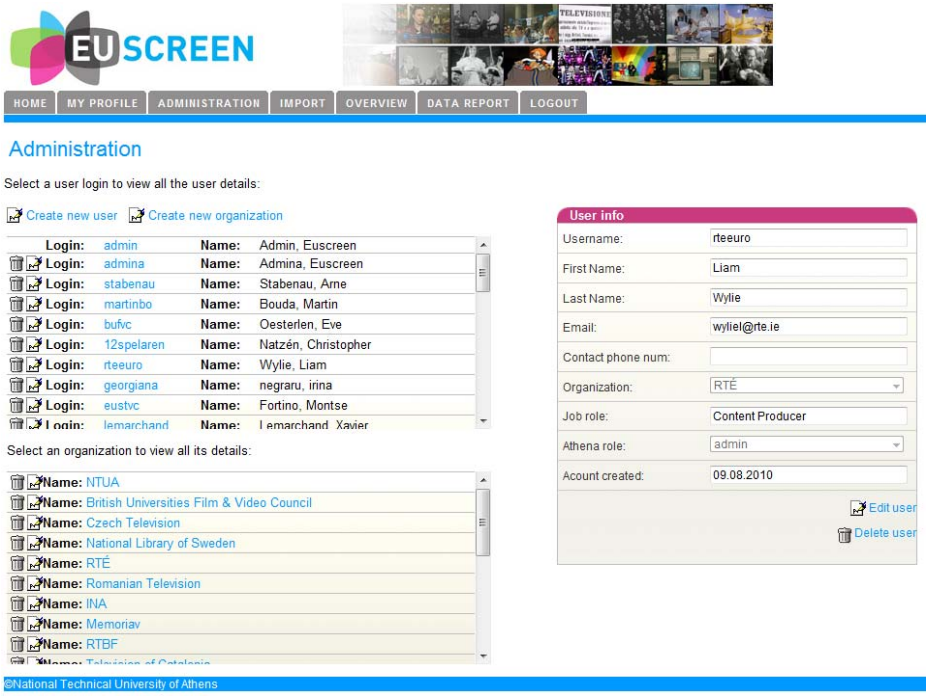


Figure 4.3 User's Administration Screen

Every user of the system has the right to see all the other users and data within the same organization (fig. 4.3). S/he can change her own details by using the Profile page (fig. 4.4).

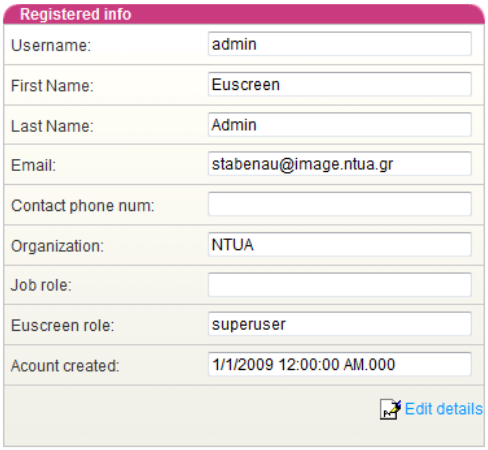


Figure 4.4 User Profile Screen

Administrators and Annotators of an organization can upload data using the import page (fig. 4.5).

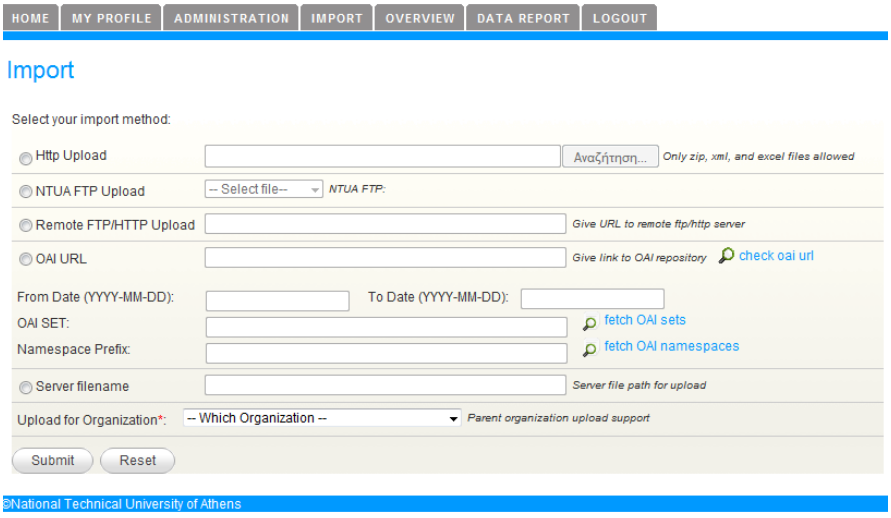


Figure 4.5 Import Interface

A history of all uploads for an organization can be browsed using the Overview interface (fig. 4.6). Different icons are used to show the current status of an import (hourglass for processing, green arrow for completed, red 'x' for failed).

An import can be deleted, thus deleting all items it contains. After an import process has been completed, mappings have to be defined for the data set in order to see all the items it contains. Every mapping defined for an organization can be saved, edited and reused at a later stage.

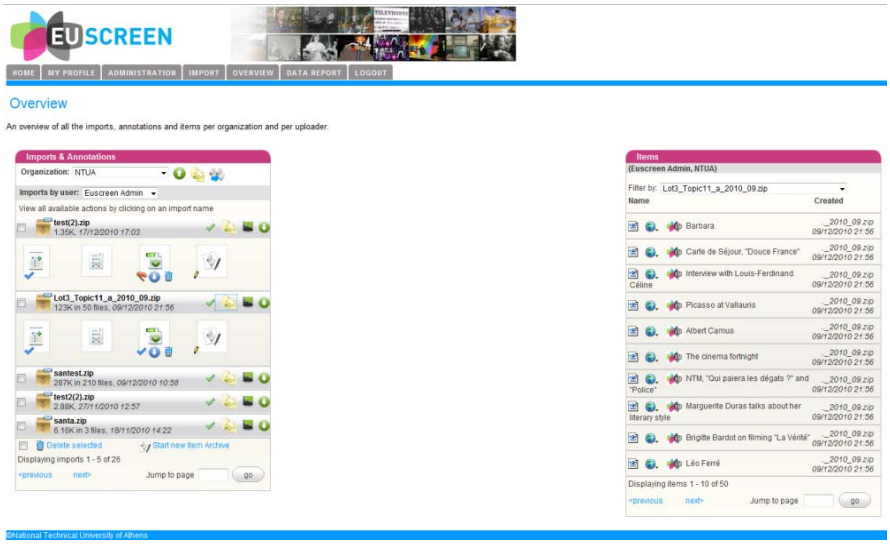


Figure 4.6 Overview Interface

In the overview screen the user can browse through items that have been uploaded. Metadata can be uploaded in a single XML containing multiple items or in multiple XML files, each one containing one item. In order to preview the items that belong to each upload, the user has to firstly define the item's root element in the XML structure (fig. 4.8) and additionally an

element that will serve to label the separated items. The actions that are available to the user as part of the Overview Interface are the following and presented in (fig. 4.7):

- When a "Green" tick appears, near the import name, it indicates that the importing process was successful. If a "Red" cross appears it means that the importing process has failed. In any case if the user hovers the mouse over the icon; various information regarding the process is displayed.
-  . This is the Show Items icon. When the user presses this button a new modal window is entered where he/she is able to review the dataset of the import. More details about this functionality in Chapter 7.
-  . This is the statistics button. When the user presses it a new modal window is rendered where various information regarding the imported dataset is presented. All the different XPaths that were extracted are presented in a tabular form together with statistical information regarding the distribution of the various values of each XPath.
-  . The download button. When the user presses it he/she is able to download an archive with all the XML files that are part of the ingested dataset.

Figure 4.7 How the imports are presented to the user in the "Overview" Tab.

When the user clicks on the name of the Import an extended view for the current Import is presented as depicted in (fig 4.9). The rest of the mandatory steps that are part of the EUscreen ingestion tool core workflow are executed from this extended view. More specifically:

-  . This button invokes the process for defining the root and label element from the extracted XPaths from the ingested data set.
-  . This button executes the transformation of the items.
-  . This button invokes the mapping tool in order to perform the semantic mappings between the source and target Schemas and produce an XSLT.
-  . This button invokes the Annotation Tool where the user is able to annotate existing items, delete them or create/add new items.

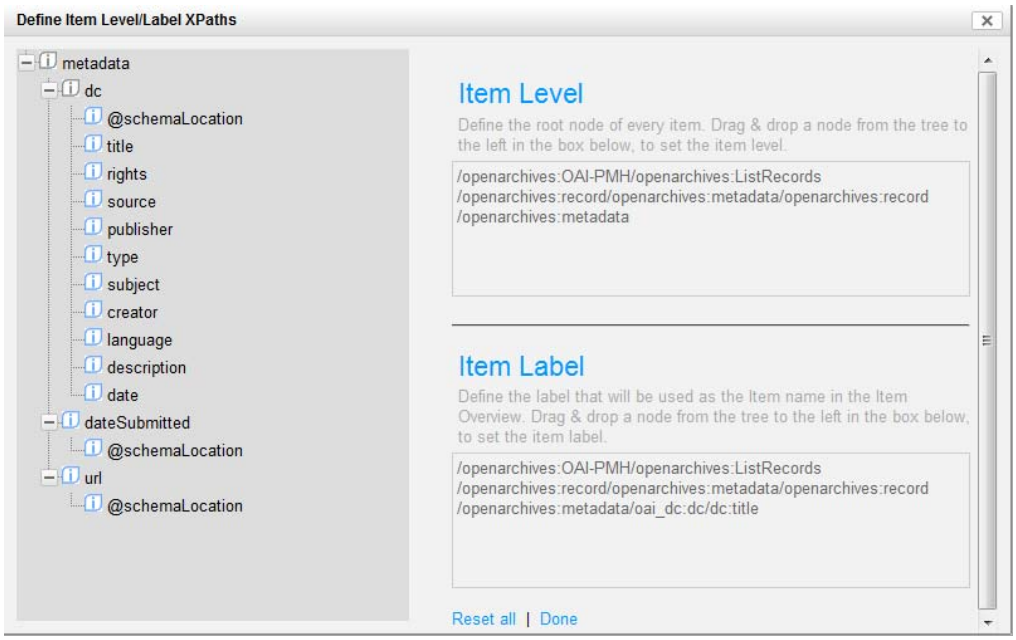


Figure 4.8 Definition of item root element

Figure 4.9 *The extended view of an Import in the "Overview" Tab.*

Having defined the item root element, the items that belong to the specific upload can now be previewed (fig. 4.10). Finally, through an icon next to each item, the user views the original XML (fig 4.11)

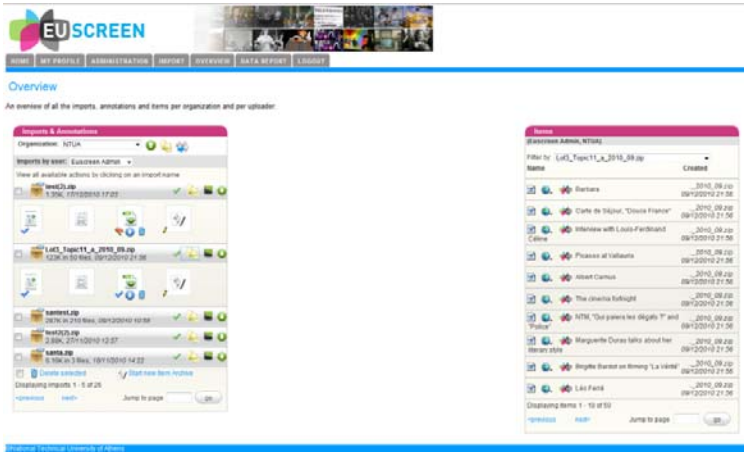


Figure 4.10 Items Screen

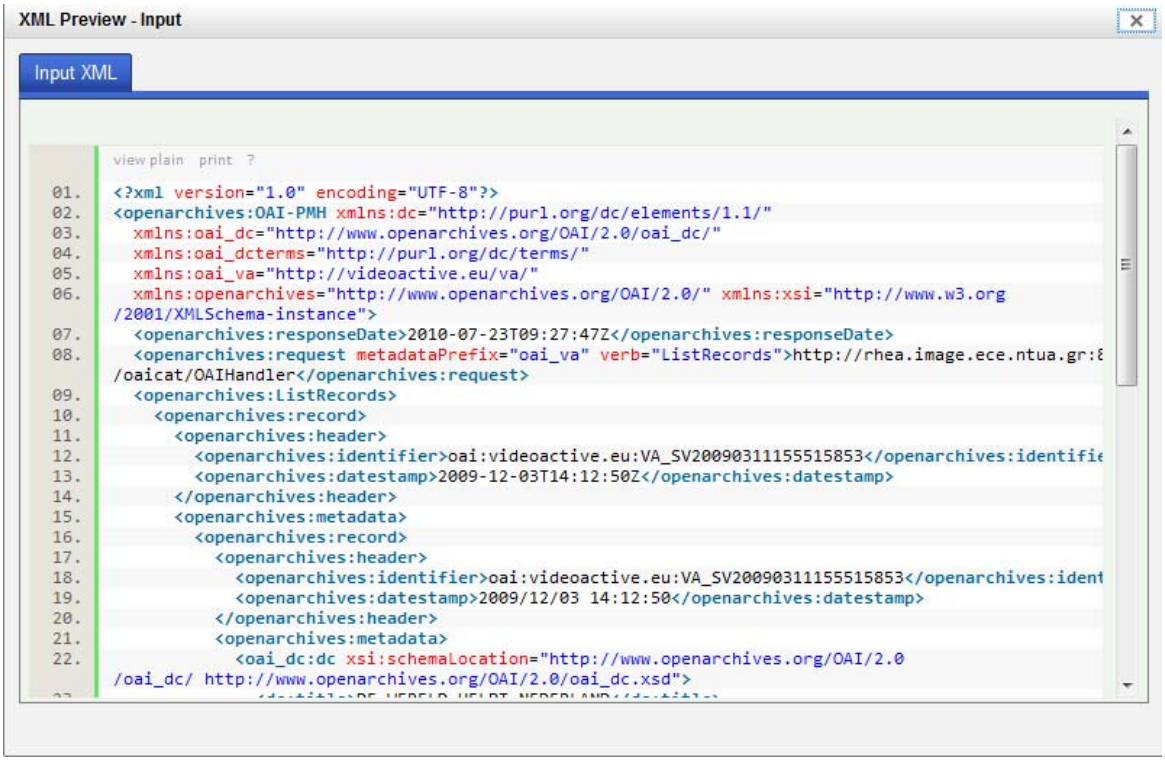


Figure 4.11 Input XML

5 Mapping module

The core module of the EUscreen web service is the mapping tool. The service shares functionality with many existing metadata repositories, e.g. DSpace and Fedora (see annex), one of its main goals it to provide support for a great diversity of metadata schemas or simple data structures, thus widening metadata interoperability. The EUscreen Ingestion platform aims to be able to store and manipulate metadata that are described using different conceptual models for encoding and decoding information. For this reason both a syntax and semantics have to be defined in order to obtain a complete and expressive model. XML is used as the machine understandable syntax and can be interpreted using different parsers depending on the specified needs. The mapping tool provides the interfaces and mechanisms for identifying and registering through a reference model the semantics of the models used.

Data integration processes comprise of various tasks including data matching, data transformation, and schema/semantic matching. Many solutions have been proposed by the community for each one of those tasks, ranging from applications that rely heavily to the user, to applications that are semi-automatic and in some cases completely automatic depending on the task, thematic category and schema complexity. For the case of schema/semantic matching many techniques and platforms have been developed, enabling the user to complete successfully the task. Notable cases are the schema mapping tool provided by Altova that offers a rich editing environment where the user is able to map any number of arbitrary schemas, but the whole process is totally manual, and the COMA++ platform that offers the user an environment for semi-automatic schema mapping, using state of the art algorithms. Europeana uses the Sip-creator tool from Delving[1].Although these approaches attempt to solve the general problem, the case of the EUscreen project poses specific requirements that lead to a more specialized solution to efficiently handle large amounts of diverse data and metadata.

By choosing a semantically rich and well-defined reference schema as the target of the mapping process the user has the opportunity to semantically enrich his data and metadata while quality of the aggregated content is ensured. The schema adopted in the EUscreen project is EUscreen Metadata Schema which is expressive enough to accommodate the project's content. The mapping process is manual and the tool offers previewing, assisting and validation capabilities in order to ensure the quality of the result. The design principles of the mapping tool ensure the extensibility of the tool itself on a software level and of the system on a data level. EBUcore is used as the storing exporting EUscreen metadata and ensuring interoperability with other systems. It can be extended to support alternative target schemas that are represented by valid XML documents.

5.1 Functional analysis

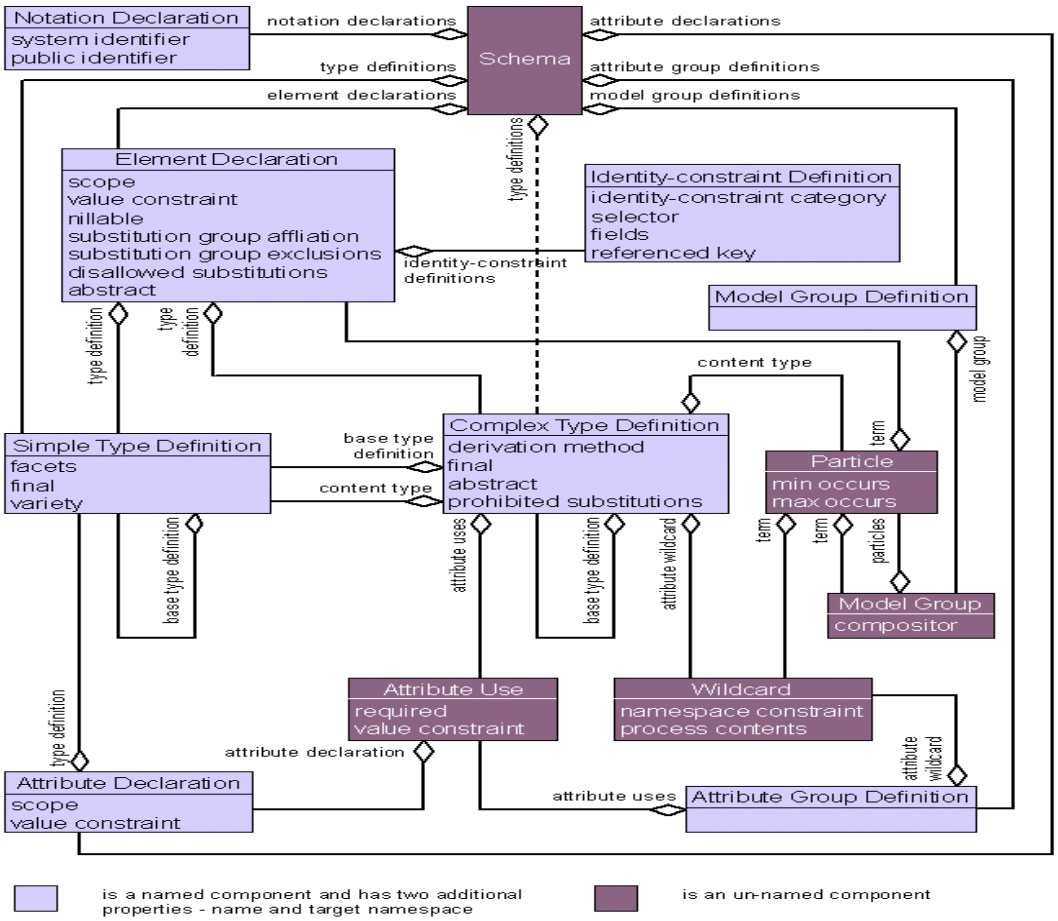
The mapping tool is designed using a client – server approach. A subset of the functionalities is implemented as server side services, while the user interface is rendered on the client inside a web browser. The communication between the client and the server is achieved using AJAX calls. One of the core design concepts of the mapping tool is that the user should be able to use all the functionality he might need in order to achieve the best possible result with

[1] https://github.com/delving/delving/tree/master/sip-creator

minimal effort. In order to achieve that, the tool must be intuitive and visual aiding and appealing. Another important design concept is that performance must be ensured because the EUscreen Ingestion platform must be able to perform computational intense tasks, e.g. metadata transformation and data parsing, without affecting the interaction between the user and the web service. This is achieved in a great degree by separating the interface rendering and the interaction with the user from intense tasks that are executed on the server side. At the same time the communication overhead between the client and the server was minimized as much as possible.

The XML Schema Parser sub-module is responsible for parsing the target XML Schema and retrieves any valuable information it's stored in its structure, e.g. annotations used for documenting the schema. After parsing the XML schema the sub-module generates an intermediate data structure serialized using the JSON language in order for the user interface to parse and generate the corresponding visual components. The rationale behind choosing JSON as the serialization language for that data structure is the software interoperability the EUscreen Ingestion platform is attempting to achieve. JSON is a well supported language with interpreters for every major language platform available which reduces the overhead introduced by using XML for exchanging messages both by a reduced memory footprint needed and a simpler structure which makes parsing a much easier and lightweight task. The XML Schema Parser sub-module requests and retrieves the schema needed from the persistent data layer. By using Hibernate for accessing and manipulating the data model, the software's architecture ensures the platform neutrality and separates the maintenance of the sub-modules from that of the data model itself.

The XML Schema parser sub-module is based on the XML Schema Object Model (XSOM) API that is part of the JAXB API for XML data binding. The main design goals of the XSOM API are a) to expose all the defined in the schema spec and b) to provide additional methods that help simplifying client applications. XSOM consists of roughly three parts; the first part is the public interface the entire functionality of XSOM is exposed by this interface to the client. The second part is the actual implementation of these interfaces. Finally the third part is a parser that reads XML representation of XML Schema and builds the XSOM data model accordingly. This part of the code is mainly generated by the RelaxNGCC API. The XML schema component model that XSOM is able to parse and represent is presented in Figure 5.1.



XML Schema Component Data Model

Figure 5.1 The XML Schema Component Data Model used with XSOM

For the needs of the EUscreen Ingestion service, an import is not required to include the schema used. This simplifies the actual work for the user and at the same time the set of schema components that have to be mapped is reduced to only those that are used, thus reducing redundancy. The Schema Generator sub-module produces the required simplified version of the schema that corresponds to a specific import by the user. When a user triggers the invocation of the mapping tool for a specific import, this sub-module is also invoked. It communicates with the data layer using the Hibernate persistent API. The next step in the workflow of the Schema Generator sub-module is to parse the data for a specific import and generate a tree like structure using HTML elements that represents the schema used. This tree like structure is then transmitted to the User Interface sub-module and is enhanced using JavaScript in order to create an interactive tree that represents a snapshot of the XML schema that the user is going to use as input for the mapping process.

The User Interface sub-module is responsible for creating and presenting an intuitive and visual appealing environment for the user to define mappings, without sacrificing any of the functionality needed to properly achieve the task of schema mapping. This sub-module is invoked by the user through the Overview interface of the EUscreen Ingestion platform per

import listed there. When the invocation occurs the server retrieves the id of the import and the workflow of the mapping tool is executed; the final step of that workflow is the transmission of all the appropriate structures to the user's browser where the mapping tool is rendered. The User Interface sub-module is implemented in JavaScript using the YUI library from Yahoo. The usage of that library for implementing the visual components also ensures cross-browser compatibility.

5.2 Mappings

The mapping tool allows the user to define semantic mappings between the source and target schemas. An XSLT is then generated, based on these mappings, that can automatically convert all imported items. An example of the mapping tool is shown in the following is depicted in (fig 5.2).

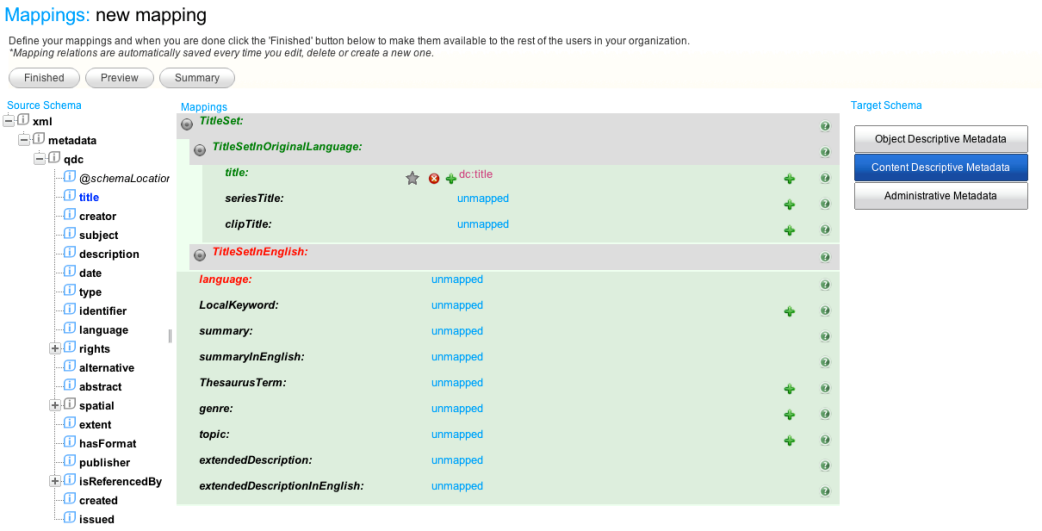


Figure 5.2 Mapping tool

5.2.1 Source schema

On the left of the mapping tool is a tree like structure of the source schema similar to the one presented during dataset item level selection. The user can navigate in the schema by clicking the nodes on the left of the tree elements. Elements with a grey information icon (ⓘ) are structural elements and do not contain data values. The rest of the elements are leaf elements with data while elements starting with '@' are attributes of the corresponding father. Elements in blue are elements that have been already used in this mapping. More information about the schema elements is provided by clicking the ⓘ icon. This action invokes a panel as show in the following Figure:

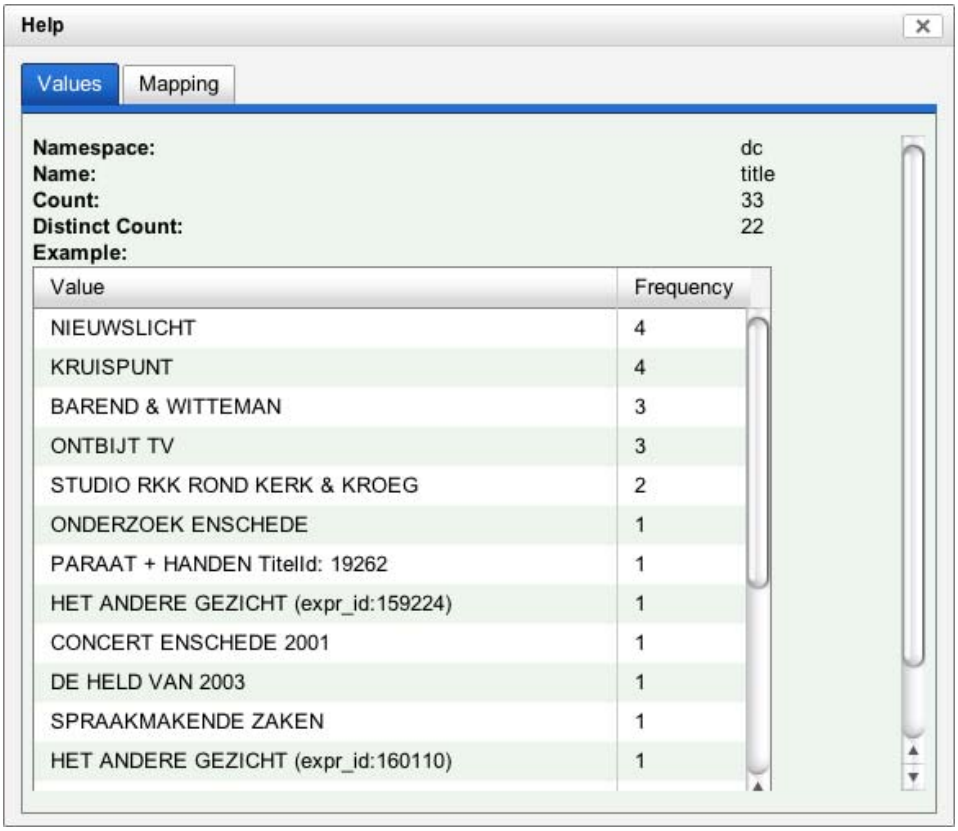


Figure 5.3 Source schema element information panel

This panel contains two tabs: Values and Mapping. The Values tab shows the following information about the specific element:

- **Namespace:** The XML namespace to which this element belongs.
- **Name:** The element name.
- **Count:** The number of times the XPath of this element exists in the imported dataset.
- **Distinct Count:** The number of unique values associated with this XPath in the imported dataset.
- **Example:** A sample of these values sorted by their frequency of appearance in the imported dataset.

The Mapping tab shows where and how the specified element is being used in the mapping that is being edited.

5.2.2 Target Schema & Mapping Area

The target XML schema is split into sections that appear as buttons on the right side of the mapping tool. These buttons are used to navigate to specific parts of the target XML schema. By clicking these buttons, the corresponding part is loaded in the middle of the mapping tool along with its specified mappings.

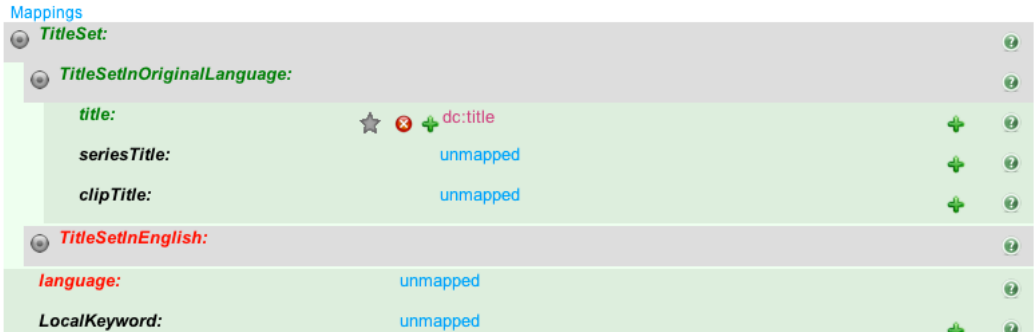


Figure 5.4 Example of the mapping definitions

Each row in the mapping area corresponds to a mapping element of the target xml schema. Rows with grey background are structural elements and contain other elements. They can be expanded and their can be seen by clicking the button to left of their name. The rest of the elements can contain data and have an 'unmapped' area which can be used to defined mappings, as explained later. Elements can also have attributes which can be seen by clicking the @ icon when available. If an element can exist more than one times in the target XML file then a + icon is available on the right of the element row. By clicking this icon an additional row will appear on the mapping area. More information about each element can be provided by clicking the corresponding icon on the right of the element row. Elements with green names are elements with mappings or have children with mappings. Elements with red names are mandatory elements that have no defined mappings or have mandatory children with no mappings defined.

5.2.3 Mapping types

Various mapping types are available by using the mapping tool. The most common type of mapping is **XPath mapping**. This type of mapping will copy data from a source element to a target element. To define this kind of mapping, drag n' drop a source element to the 'unmapped' area of a target element. The source element will then appear in the unmapped area of the target element. By clicking the + icon on the left of the source element name, an additional unmapped area will appear for the same target element. More mappings can be performed on this unmapped area and the XML result will contain a concatenation of the provided values. Clicking the corresponding icon will remove a defined mapping.

Double clicking on the unmapped area will define a **constant value mapping**. The following panel is invoked:

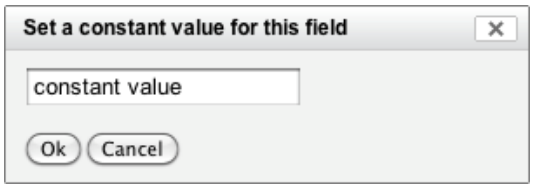


Figure 5.5 Constant Value Panel

The user can type a constant value in the provided text field. The value will then appear in the mapping area and in the result XML files. This type of mapping is useful for text that is intended to appear in all transformed items. Constant value mappings can be combined with XPath mappings to construct specific values such as URLs.

5.2.4 Conditions

Mappings can be restricted so that they will apply only under certain conditions. To define these conditions the ★ button is used. This will allow the input of condition as shown in the following Figure:

Figure 5.6 Mapping Condition

The conditions supported are in the form of <Source XPath> = <Constant Value>. The corresponding mapping will apply only if the source XPath data for a specific item equals to the constant value provided. The condition source XPath can be set by dragging n' dropping a source element to the condition input area as shown in the previews Figure. The constant value can be set by double clicking on the constant value area.

If a more complex condition is required then the provided condition editor must be used by clicking on the ⊖ icon next to the condition. An example of the condition editor is shown in the following figure:

Figure 5.7 Condition Editor

The condition editor provides the means to define more complex conditions. In addition to each clause's source element and constant value, the relational operator can be set. The logical operator that combines the clauses can also be defined. Additional clauses or subclauses can also be created as needed, by clicking the corresponding ✚ icon.

5.2.5 Preview & Mapping Summary

A summary of the defined mappings can be seen by clicking the 'Summary' button on the top of the mapping tool. This will invoke the following panel:

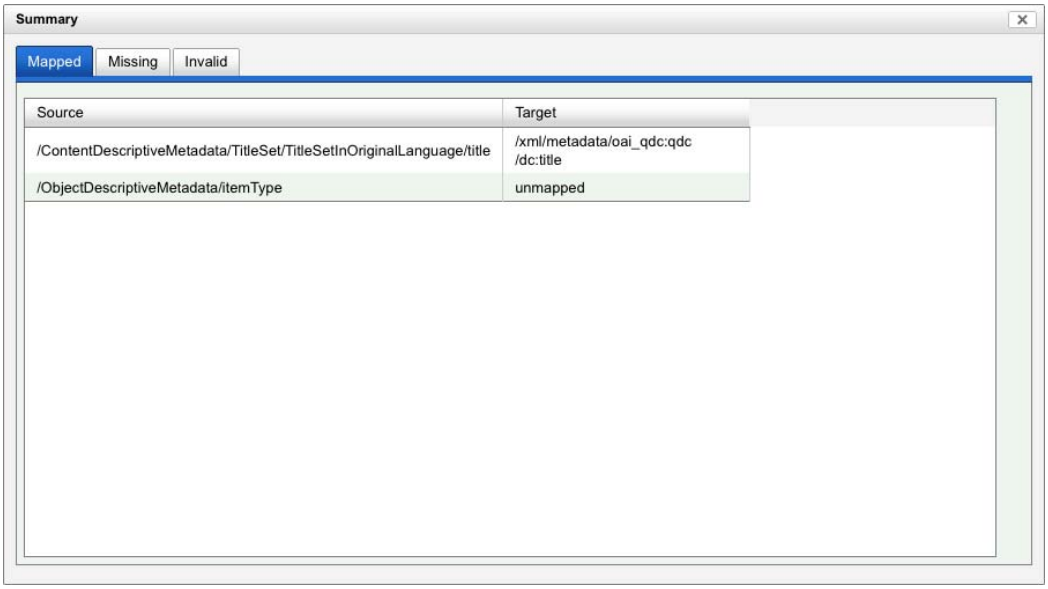


Figure 5.8 Mapping Summary Panel

This panel contains the following tabs:

- **Mapped:** All mapped source elements and the corresponding target elements.
- **Missing:** Mandatory target elements that have no mappings.
- **Invalid:** If a mapping definition was loaded based on another dataset, all XPaths form this dataset that do not exist in the current dataset appear in this tab.

The generated XSL can be previewed at any time by clicking the 'Preview' button on the top of the mapping tool. This will invoke the following panel:

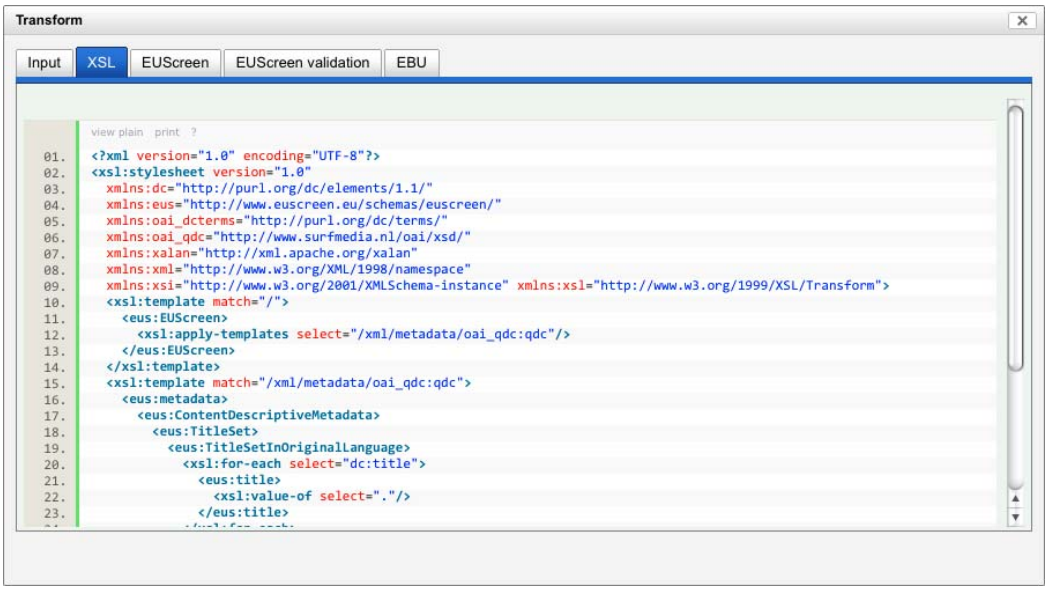


Figure 5.9 Preview Transform Panel

This panel contains the following tabs:

- **Input:** The XML that corresponds to the first item from the imported dataset.
- **XSL:** The generated XSL.
- **EUscreen:** The item transformed to the EUscreen schema based on the previous XSL
- **EUscreen:** Validation information for the transformed item. Any errors from wrong or incomplete mappings will be reported there.
- **EBU:** The EUscreen item transformed to the EBU schema.

5.3 Software Interfaces

The mapping tool of provides and requests interfaces from various other modules of the EUscreen Ingestion platform. More specifically there are the following interfaces:

- Interface from the XML Schema Parser sub-module to the User Interface sub-module.
- Interface from the XML Schema generator sub-module to the User Interface sub-module.
- Interface from the User Interface sub-module to the Item Summarization module.
- Interface from the EUscreen Ingestion platform Data Layer to the User Interface sub-module.
- Interface from the EUscreen Ingestion platform Data Layer to the XML Schema Parser sub-module.
- Interface from the EUscreen Ingestion platform Data Layer to the XML Schema Generator sub-module.

Interface Name	Interface from the EUscreen Ingestion platform Data Layer Module
Participant Module/Component	EUscreen mapping tool UI sub-module
Participant Module/Component	EUscreen Ingestion platform Data Layer
Invoked/Invokable Methods	
Method	Description and Exchange Format/Type
Retrieves or saves the mappings defined by the user so far.	The UI sub-module communicates with the EUscreen Ingestion platform Data Layer and stores or retrieves the mappings for a specific import. **Input:** A JSON representation of the mappings or the id of an import. **Output:** A status code or a JSON representation of the mappings of an import.

Interface Name	Interface from the EUscreen Ingestion platform Data Layer Module

Participant Module/Component	XML Schema Parser sub-module
Participant Module/Component	EUscreen Ingestion platform Data Layer
Invoked/Invokable Methods	
Method	Description and Exchange Format/Type
Retrieves the target XML Schema.	The XML Schema Parser requests the target XML schema in order to parse it and generate the appropriate data structures. **Input:** none. **Output:** The target XML Schema.

Interface Name	Interface from the EUscreen Ingestion platform Data Layer Module
Participant Module/Component	XML Schema Generator sub-module
Participant Module/Component	EUscreen Ingestion platform Data Layer
Invoked/Invokable Methods	
Method	Description and Exchange Format/Type
Retrieves the data of a specified import in order to generate a tree representation of the Schema.	The XML Schema Generator requests the data of a specific import in order to generate a tree representation of the Schema. **Input:** The id of an import. **Output:** The data of that import.

Interface Name	Interface from the EUscreen Ingestion platform mapping tool UI sub-module
Participant Module/Component	Item Summarization module
Participant Module/Component	EUscreen Ingestion platform mapping tool UI sub-module
Invoked/Invokable Methods	
Method	Description and Exchange Format/Type

Invokes the mapping tool for a specific import.	The user is able through the item summarization module to invoke the EUscreen mapping tool for a specific import. **Input:** The id of an import. **Output:** none.

Interface Name	Interface from the Schema parser sub-module
Participant Module/Component	EUscreen Ingestion platform mapping tool UI sub-module
Participant Module/Component	EUscreen Ingestion platform mapping tool Schema parser sub-module
Invoked/Invokable Methods	
Method	Description and Exchange Format/Type
Requests a JSON representation of the target Schema that is going to be rendered on the UI.	The UI sub-module requests from the Schema parser sub-module a JSON representation of the target XML Schema. **Input:** none. **Output:** A JSON representation of the target XML Schema.

Interface Name	Interface from the Schema generator sub-module
Participant Module/Component	EUscreen Ingestion platform mapping tool UI sub-module
Participant Module/Component	EUscreen Ingestion platform mapping tool Schema generator sub-module
Invoked/Invokable Methods	
Method	Description and Exchange Format/Type
Requests a tree representation using HTML of the generated XML Schema for a specific import.	The UI sub-module requests from the Schema generator sub-module a tree representation of the Schema generated for a specific import. **Input:** The id of a specific import. **Output:** A tree representation of the generated XML Schema in HTML markup for the requested import.

6 Statistics

The EUscreen Ingestion platform aims to be able to handle millions of metadata records that will be represented using the XML language. For many reasons the necessity of an easy way to inspect these metadata arises, mainly because it will be impossible for a user to inspect every single record without spending either too much time or making unavoidable mistakes. Also for many procedures that are part of the EUscreen Ingestion platform workflow, e.g. quality assurance (QA) procedure, it is mandatory to design and implement a module that will be able to extract valuable information in the form of statistics from the imported metadata datasets. The basic design goal of the statistics tool is to implement a set of functions or methods that will be applied to the ingested metadata datasets. These methods will produce a set of metrics that will be valuable for the user in order to survey reliably and fast the metadata he wishes and also for the machine to be able to exploit the generated information in order to optimize various procedures that are part of the EUscreen Ingestion platform workflow. In order to rationalize the decisions made for the design of the Statistics tool we have to describe the nature of the data we are attempting to analyze.

As it is mentioned above, the platform is going to store, handle and manipulate structured data and metadata that are described using the XML language. The XML Specification describes both syntax and a model. The syntax is based on angle brackets, while the model is basically a tree of nodes, which could be either elements, attributes or text containers. The XML model is described in detail in the XML Infoset specification provided by the W3C.

The ‘names’ of those nodes are not dictated by the XML model or syntax. XML is in fact a meta-language, a language to describe languages. In general there is no reason why nodes with the same name or content should be reused throughout the same XML tree, but in real life the number of different ‘names’ for those nodes is very small compared to the amount of data that the XML tree contains. In practice, the amount of different ‘names’ is a function of the different namespaces used by the XML tree and not a function of the size of the XML tree itself.

This is a basic characteristic of the XML language as it is used, upon which the generation of statistics makes sense while at the same time it provides valuable information on how to organize the generation of the statistics and which metrics should be used, as it is going to be presented in more detail later in this document. One example that clarifies the above statement is the following; the web contains billions of documents, and it is possible to convert all of them into XHTML and create a huge XML dataset several Terabytes big. Despite the size of this dataset, the structural complexity of that XML dataset would not be a function of its size, but a function of the schemas used, mainly XHTML.

Another interesting property that real life XML datasets exhibit is the fact that ‘semantic’ information (information that shows the user how to understand this) is not encoded in the node name but in the entire sequence of names that leads to that node. In brevity, in its XPath. An example of this is the use of the node “email” that has no meaning by itself, if not indicating that it contains an email address, but the path fragment “person/email” will indicate that this “email” node is contained inside the “person” node and for that reason, is associated

to that. Real life analysis shows that the number of XPaths is a function of the complexity of the schema, therefore, again, not a function of the size of the dataset.
Apart from the above mentioned characteristics, in order to be able to start specifying the set of the methods and metrics needed to produce a complete and useful statistics for XML data, someone also has to make a few assumptions. The assumptions made in the case of the EUScreen Ingestion platform are the following:

- The data is encoded in literals, either as textual nodes inside elements or as attributes values.
- The metadata is encoded in XPaths that identify those literals.
- Mixed content is considered as data.

Given those assumptions and the elements of the analysis of the basic characteristics of XML so far, we would like to be able to "inspect" the dataset and generate statistics that derive from the answers of the followings questions:

- What XPaths were used in this dataset?
- How many times were they used?
- What is the distribution of the values a specific element has?
- How many different schemas are used in a specific dataset?

The EUScreen Ingestion platform follows a complete workflow for ingesting, managing, transforming and exposing metadata in various formats. For this reason the statistics that are useful are not only those related to the XML data. For this reason we need to define various levels of statistics that should be generated. These different levels are the following,

- Collection related statistics.
- XML Element statistics.
- Value Statistics.

Collection related statistics are those related to the ingestion process itself. For example, the number of distinct XML documents a specific user has ingested so far. XML Element statistics are those related to the questions stated earlier on this document. Finally, the value statistics refer mainly to the distribution of the various values of a specific XML element or attribute. Every set of statistics come in response to specific design needs of the workflow. The collection statistics for example, are needed for versioning and quality control. The XML element statistics are needed in the process of mapping and for inspecting the quality of the ingested metadata manually. Finally the value statistics are needed for ensuring the quality of the ingested metadata and provides an easy way to pinpoint errors on data, mainly on those controlled by specific vocabularies, but also needed in the process of normalization and enrichment.

6.1 Functional analysis

One of the core design concepts of the statistics module is the readability and accessibility of the various statistics in a graphical appealing and intuitive way. From a software engineering perspective the architecture of the statistics module is based on principles that ensure reusability and scalability of the generated application.

The statistics module is separated in two distinct sub-modules. The first one is responsible for the generation of the statistics and the second one is responsible for the visual presentation of the statistics to the user. The user interface sub-module of the statistics module is written

using JavaScript and is an application rendered and managed completely by the clients' browser. The generation of statistics sub-module runs on the server side and is integrated with the data layer. A proper interface is defined between the two sub-modules in order to maintain communication; this interface is based on AJAX calls made by the client to the server asynchronously. The rationale behind this architecture design is to decouple as much as possible the processing that happens to the server from the presentation part of the statistics module. In this way, the scalability and responsiveness of the whole system is ensured, especially when these principles are maintained in every step of the EUscreen Ingestion platform workflow and the various applications that are defined in each one. Another advantage of this architecture is that it is possible to separate the implementation effort of the UI part and the core functionality that is needed. A consequence of this is that many changes can happen on the UI part in a consistent way, without affecting any other module or functionality of the system, so, also the extensibility of the statistics module is ensured.

The statistics generation sub-module is mainly an object that implements a specific interface for requesting the required data from the database layer and calculate the appropriate statistics based on them. For this reason a native persistent API is used based on Hibernate in order to be able to access the EUscreen Ingestion platform Data Layer and execute queries that will produce the requested statistics. The reason behind choosing a persistent technology based on Hibernate is to ensure platform neutrality from the various databases and schemas that could be used for storing data internally. By ensuring platform neutrality, also the extensibility of the statistics module is ensured.

The User Interface sub-module is responsible to present to the user information from very large datasets in a consistent and visual appealing way. The user invokes the statistics module from the item summarization tab which is part of the web service interface and where the user is able to manage his imported metadata. Each import has a distinct button which when pressed transfers the user to a new screen (Figure 6.1) where the statistics user interface is rendered and the user is able to start inspecting the statistics of that particular import. A minimal set of information is transferred to the server when the user invokes the statistics tool; this information consists mainly of a unique id that identifies the specific import so the invoked module is able to load any data that are needed in order for the statistics to be calculated from the data layer.

When the user invokes the statistics module for a specific import, he is presented with the first set of statistics that refer mainly to the Schema and the XML Element information that can be calculated for that import. The screen is separated in two distinct areas. The first one from the right is used to present to the user the various XML schemas and prefixes for each of those that are present in this particular import. The second area of the first page consists of a view separated with tabs, for each XML Schema presented on the table that was described previously, also one tab is created that holds a table with information regarding the elements found in the specific import and that belong to that Schema. This table has 5 rows where statistics are presented, the names of the columns and their roles are the following:

- **Element.** This is the name of the Element or the attribute of an element found in the import and that belongs to a specific XML Schema.

- **Frequency.** This is the frequency of a specific element compared to the number of distinct items that are present in the import.

- **Unique.** The numbers of distinct/unique values the element or attribute holds.
- **Length.** The average length of the values the element or attribute holds.
- **Sparkline.** A visual representation in the form of a graph that has the same size with the text font used in the table. This is used mainly to provide to the user a fast and visual way of understanding the distribution of the element or attribute in the import.

The user is able to distingue the elements from their corresponding attributes although both are presented in different rows of the table. This is achieved by positioning differently the elements from the attributes while grouping them together. The attributes of a specific element, always follow it in successive rows while an attribute always has the character @ in front of its name. Another visual aid of the statistics page is the usage of different colors that correspond to different characteristics that were found related to the data. These color codes together with their meaning are the following:

- Green. An element or an attribute has this color when the value of frequency equals the value of uniqueness. This characteristic might indicate a unique element or attribute that is or could be used as a possible identifier.
- Red. An element or an attribute has this color when every occurrence of it in the import had an empty string as a value. In this way the user is able to quickly identify any elements that are empty but they shouldn't be.
- Grey. An element or attribute has this color when it is of type ComplexType. This indicates elements that should not hold any value but are wrappers of others.
- Blue. An element or an attribute has this color when none of the above criteria are present for them. Usually the majority of the elements that are supposed to hold a value should have this color code.

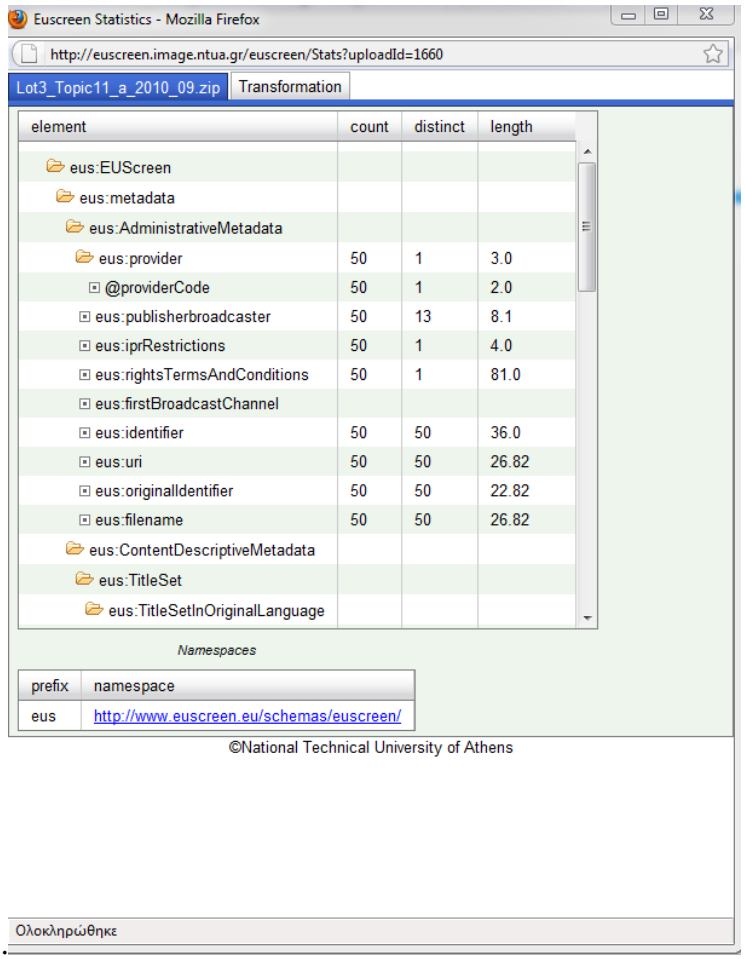


Figure 6.1 Input schema statistics.

Another important set of statistics is related to the values of a specific element or attribute. These statistics are mainly used for the user to be able to inspect the content of a specific element or attribute and is only applicable for certain cases where the length of the values do not exceed a certain threshold that is defined. This threshold exists for two reasons. The first reason is that while the median length of the value increases, so do the chances that the set of value lacks any coherency, so by presenting all the values would result in a huge view of data while not offering any valuable information to user. The second reason is to reduce the computational overhead on the server. Also for optimizing the performance both on the client and the server an interface based on AJAX calls is implemented that retrieves the data related to a specific element or attribute when the user double clicks on one of them as they are presented on the first page of the statistics user interface sub-module. In this way it is possible to reduce the memory footprint on the client side while at the same time reduce the computational overhead of calculating statistics for every possible element or attribute, on the server side. When the user wishes to view statistics related to the values of a specific element or attribute a modal window pops up separated in two distinct parts (Figure 6.2). The right part contains an HTML table enhanced with functionality using JavaScript using the YUI library that has the following two columns.

- **Value**. The rows under this column contain a distinct value found for a specific element or attribute.

- **Frequency**. The rows under this column contain the frequency of that specific value that appears in the preceding row.

The left part of the modal window presents the data of the table in a visual way using a pie chart. In the future the user will be able to select dynamically the type of diagram he wishes to view, e.g. bar diagram instead of a pie chart. When the user decides that he finished inspecting the value distribution of a specific element or attribute he is able to close the modal window and return to the first view of the statistics User Interface sub-module. All the data are retrieved from the server using AJAX calls and rendered in the client side for performance reasons.

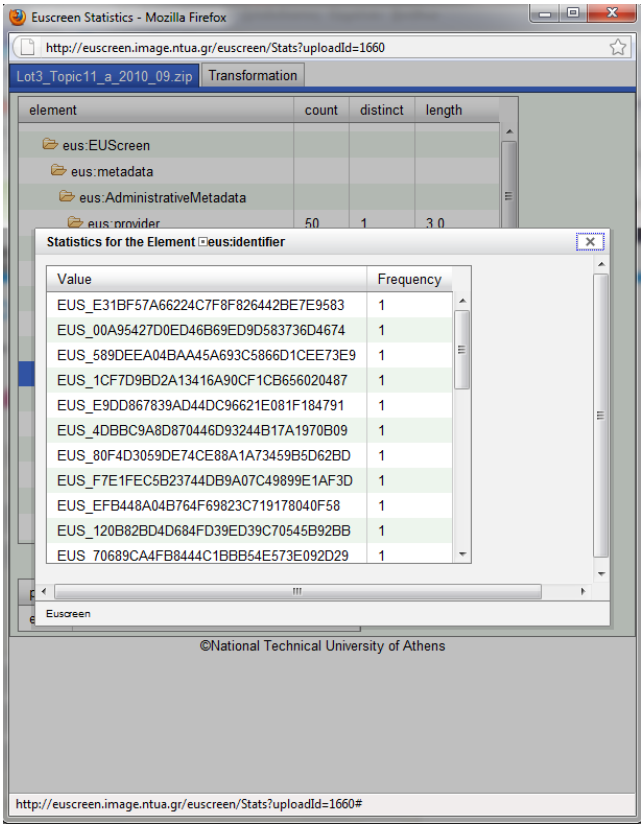


Figure 6.2 Value distribution statistics for a specific element or attribute.

6.2 Software Interfaces

The statistics module of the EUScreen Ingestion platform provides and requests interfaces from various other modules of the platform. More specifically there are the following interfaces:

- Interface from the EUscreen Ingestion platform Data Layer Module.

- Interface to the EUscreen Ingestion platform Item Summarization Module.
- Interface to the EUscreen Ingestion platform Mapping Tool Module.

Interface Name	Interface from the EUscreen Ingestion platform Data Layer Module
Participant Module/Component	EUscreen Ingestion platform Statistics Module
Participant Module/Component	EUscreen Ingestion platform Data Layer
Invoked/Invokable Methods	
Method	Description and Exchange Format/Type
Retrieves a set of pre-calculated statistics as requested by the EUscreen Ingestion platform Statistics Module	The EUscreen Ingestion platform data layer module executes a set of queries and returns a set of statistics as requested by the EUscreen Statistics module. **Input:** the name of an element, attribute, namespace or import. **Output:** Statistics generated by the execution of the appropriate queries on the EUscreen Ingestion platform database.

Interface Name	Interface to the EUscreen Ingestion platform Item Summarization Module
Participant Module/Component	EUscreen Ingestion platform Item Summarization Module
Participant Module/Component	EUscreen Ingestion platform Statistics Module
Invoked/Invokable Methods	
Method	Description and Exchange Format/Type
The user is able from the Item Summarization Module to invoke the EUscreen Ingestion platform Statistics Module in order to be presented with the statistics for a specific import.	The EUscreen Ingestion platform Statistics Module is invoked by the user from the Item Summarization Module for a specific import. **Input:** The ID of a specific import. **Output:** A set statistics for that import.

Interface Name	Interface to the EUscreen Ingestion platform Mapping Tool Module
Participant Module/Component	EUscreen Ingestion platform Mapping Tool Module
Participant Module/Component	EUscreen Ingestion platform Statistics Module
Invoked/Invokable Methods	
Method	Description and Exchange Format/Type
The EUscreen Ingestion platform Mapping tools is able to retrieve a minimal set of statistics useful for the procedure of mapping.	The EUscreen Ingestion platform Mapping tool requests a minimal set of statistics for a specific element from the EUscreen Ingestion platform Statistics module in order to assist the user in the process of mapping his schema to the EUscreen Metadata Schema. **Input:** The name of a specific element or attribute. **Output:** A minimal set of statistics for this particular element or attribute.

7 Transformation Services

7.1 Transform

When the user has successfully defined the root and label elements for a specific “Import” and the mappings between the extracted source XML Schema and the target Schema of the system, he/she is able to perform the transformation of the data. For this to happen the user has to click the  button which is visible under the extended view of a specific “Import” in the “Overview” tab. When this event is triggered by the user, he/she is presented with the modal window depicted in (fig. 7.1). The user is presented with information regarding the different states a mapping might be based on the appropriate Icons. In order for the user to continue the transformation process he/she has to select a mapping from the drop down list and click on the “Submit” button.

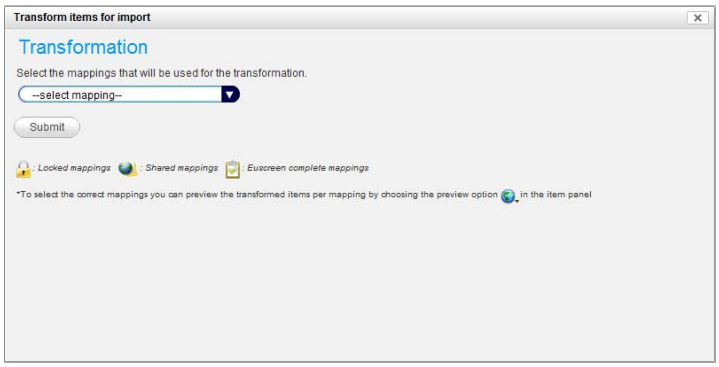


Figure 7.1 The transformation modal window.

In the case where the mapping is not correct, for example mandatory mappings are missing, the user is presented with a modal window explaining what the problems are, like the one depicted in (fig. 7.2). This modal window has two distinct tabs presenting different kind of information to the user. The first tab presents any missing mappings to mandatory XPaths of the target Schema while the second one presents XPaths with erroneous mappings. The user is able to review the errors and then he/she has to go back and either select a different mapping or complete/correct the current select one.

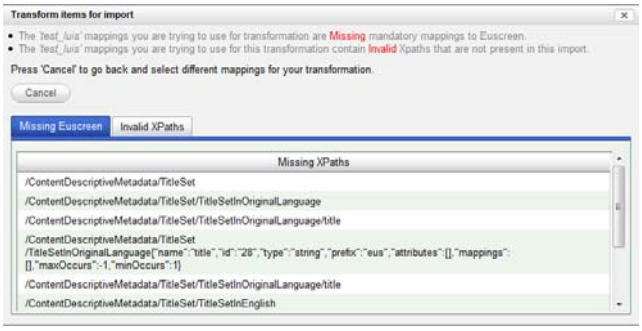


Figure 7.2 the modal window presenting the errors found in a selected for transformation mapping.

If everything goes well and the defined mappings do not contain any invalid XPaths or and there are no Missing mandatory mappings the user is redirected back to the "Overview" tab and an animated Icon takes the position of the button. When the user positions the mouse pointer over the animated icon, information regarding the progress of the transformation process is presented to him on a tooltip. Actually, while in the process of transformation, the system extracts each item XML instance based on the root element the user has defined and applies the XSLT transformation that is generated through the process of defining the mappings between the two Schemata, every generated item is then stored to the EUscreen ingestion tool persistent data layer and is associated with the current Import. In the case where an error occurs in the process of transformation a "Red X" icon appears on top of the icon. When the user positions the mouse pointer on top of the icon he/she is able to review the errors that caused the transformation process to abort. In the case where the transformation ends without errors a "Blue" tick sign appears on top of the "transformation" icon and the user is able if he/she wishes to download the transformed items.

7.2 Review Transformed Dataset

After the completion of the transformation step in the EUscreen ingestion tool core workflow, the user is able to review the original data together with the resulted transformed items and the generated XSLT on a per item level through the item browser in the "Review" tab. In order to do that the user has to press the for an individual item in the item browser. When this event occurs the user is prompted with a modal window where he/she is able to review the results of the whole process as depicted in (fig. 7.3). The user is able to view the Input XML, the Invalid XPaths if any exist, the XSL generated in the mapping process, the output XML in the target Schema of the system and the output XML of the Publishing Schema.

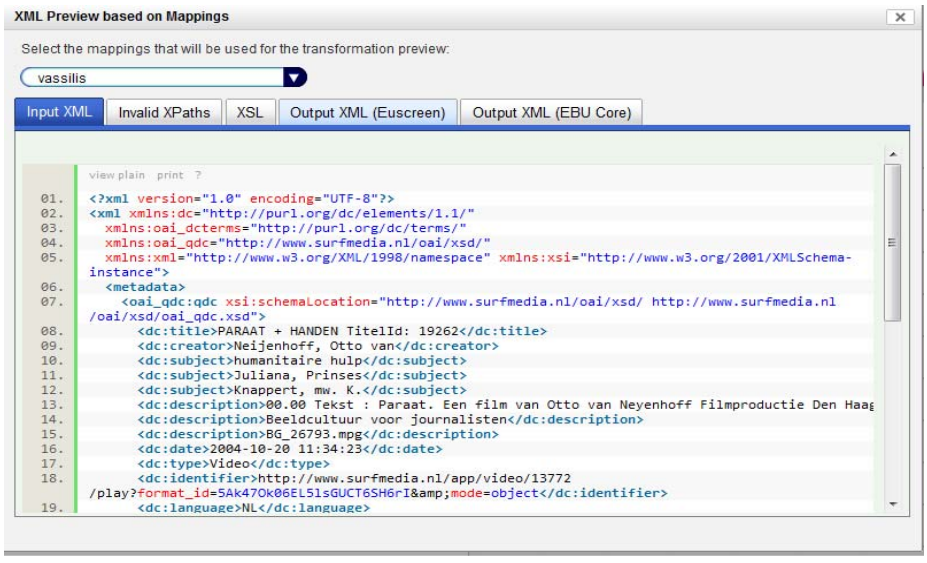


Figure 7.3 The modal window where the user is able to review the results of a transformation on an individual item.

8 Annotation Services

An important functionality of the EUscreen ingestion tool is the Annotation Process. In many cases the source XML Schema which is derived from the imported dataset, does not hold all the necessary information to fill the mandatory fields of the target Schema. Also the user might want to create new items, directly in the EUscreen Schema. For those two reasons the EUscreen ingestion tool provides the Annotation Functionality of an imported and transformed dataset or the creation of a new one where the user can create and manage new items from scratch.

When the user wishes to create a new dataset from scratch, he/she can invoke this functionality by pressing the «*Start New Annotation*» button in the Overview menu, as it is depicted in
Figure 8.1 The overview panel where the user can also access the Annotation Functionality. When the user clicks on the «Start New Annotation» button a modal window appears on which he/she is prompted to provide a name for the annotation, as depicted in Figure . After pressing on the «Done» button, a new item will appear on the «Imports & Annotations» overview and the user will be able to click on the appropriate «Annotation» and be redirected to the Annotation Tool where he/she will be able to create and annotate new items for the predefined dataset.

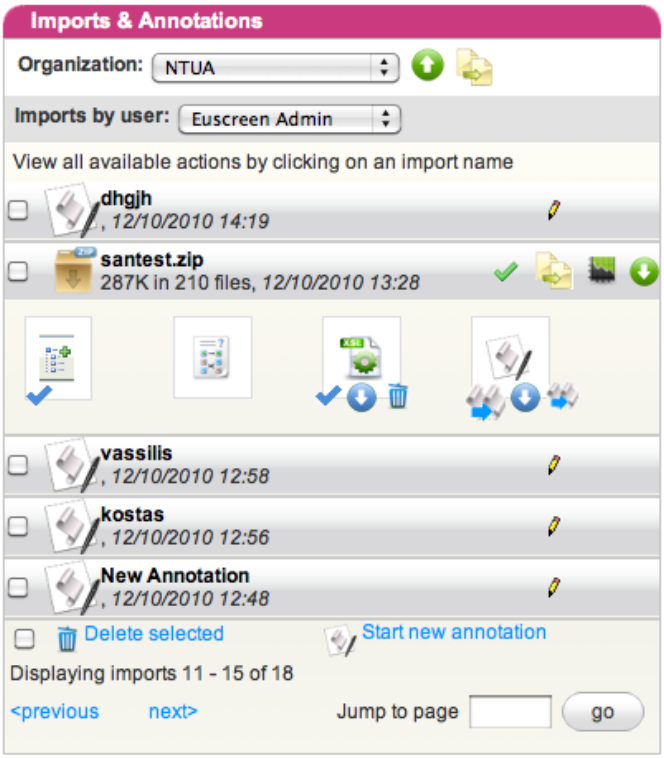


Figure 8.1 The overview panel where the user can also access the Annotation Functionality.

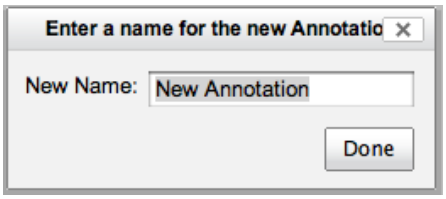
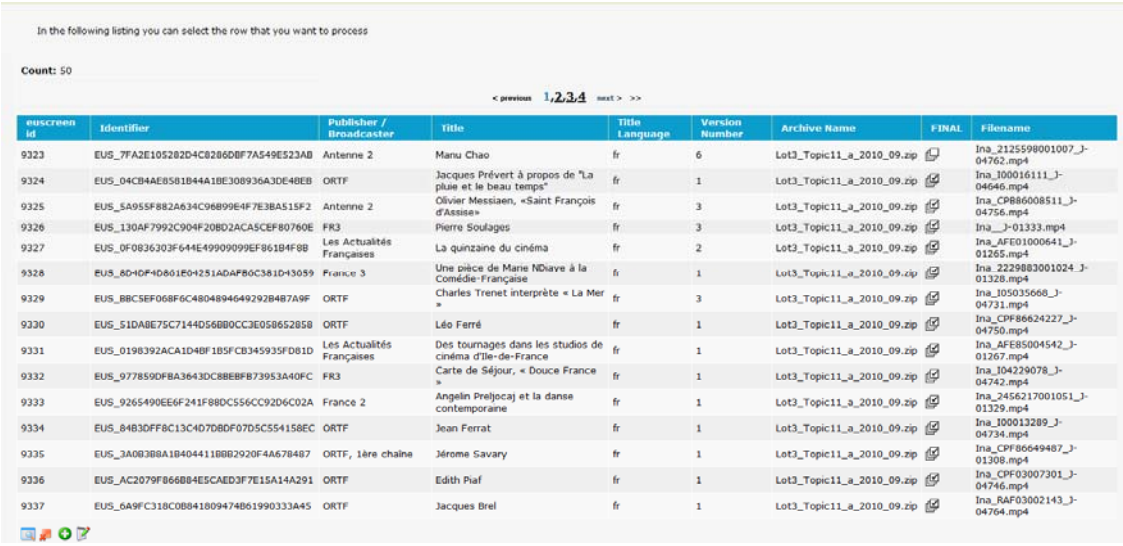


Figure 8.2 The modal window where the user types the name of the new annotation.

The second case is where the user has ingested a dataset and after a successful transformation he/she desires to view the items in the Annotation Tool and use its functionalities, e.g. create or delete items and edit the values of the already transformed ones. In this case the user has to click on the Import he/she desires in order to expand and view the available options on the «Imports & Annotation» window. By pressing the  the user will be redirected to the Annotation Tool where all his/her items will be presented and by using the available functionalities additions, deletions and modifications of the transformed items can be executed, as it is depicted in (fig. 8.3). The items are presented in a grid view where a subset of metadata is presented in order to make the whole process of choosing the appropriate item easier. Finally when the user has finished annotating the dataset on the Annotation Tool, all the changes are also reflected on the available functionalities of the «Imports & Annotations» window of the EUscreen ingestion tool.


In the following listing you can select the row that you want to process

Count: 50

< previous 1,2,3,4 next > >>

euscreen id	Identifier	Publisher / Broadcaster	Title	Title Language	Version Number	Archive Name	FINAL	Filename
9323	EUS_7FA2E105282D4C8286DBF7A549E523AB	Antenne 2	Manu Chao	fr	6	Lot3_Topic11_a_2010_09.zip		Ina_2125598001007_J-04762.mp4
9324	EUS_04CB4AE8581B44A1BE308936A3DE48EB	ORTF	Jacques Prévert à propos de "La pluie et le beau temps"	fr	1	Lot3_Topic11_a_2010_09.zip		Ina_I00016111_J-04646.mp4
9325	EUS_5A955F882A634C96B99E4F7E3BA515F2	Antenne 2	Olivier Messiaen, «Saint François d'Assise»	fr	3	Lot3_Topic11_a_2010_09.zip		Ina_CPB86008511_J-04756.mp4
9326	EUS_130AF7992C904F208D2ACA5CEF80760E	FR3	Pierre Soulages	fr	3	Lot3_Topic11_a_2010_09.zip		Ina__J-01333.mp4
9327	EUS_0F0836303F644E49909099EF861B4F8B	Les Actualités Françaises	La quinzaine du cinéma	fr	2	Lot3_Topic11_a_2010_09.zip		Ina_AFE01000641_J-01265.mp4
9328	EUS_8D4DF4D861E042514DAFB6C381D43059	France 3	Une pièce de Marie NDiaye à la Comédie-Française	fr	1	Lot3_Topic11_a_2010_09.zip		Ina_2229883001024_J-01328.mp4
9329	EUS_BBC5EF068F6C48048946492928B4B7A9F	ORTF	Charles Trenet interprète « La Mer »	fr	3	Lot3_Topic11_a_2010_09.zip		Ina_I05035668_J-04731.mp4
9330	EUS_51DA8E75C7144D56B80CC3E058652858	ORTF	Léo Ferré	fr	1	Lot3_Topic11_a_2010_09.zip		Ina_CPF86624227_J-04750.mp4
9331	EUS_0198392ACA1D4BF1B5FCB345935FD81D	Les Actualités Françaises	Des tournages dans les studios de cinéma d'Ile-de-France	fr	1	Lot3_Topic11_a_2010_09.zip		Ina_AFE85004542_J-01267.mp4
9332	EUS_977859DFBA3643DC8BEBFB73953A40FC	FR3	Carte de Séjour, « Douce France »	fr	1	Lot3_Topic11_a_2010_09.zip		Ina_I04229078_J-04742.mp4
9333	EUS_9265490EE6F241F88DC556CC92D6C02A	France 2	Angelin Preljocaj et la danse contemporaine	fr	1	Lot3_Topic11_a_2010_09.zip		Ina_2456217001051_J-01329.mp4
9334	EUS_84B3DFF8C13C4D7DBDF07D5C554158EC	ORTF	Jean Ferrat	fr	1	Lot3_Topic11_a_2010_09.zip		Ina_I00013289_J-04734.mp4
9335	EUS_3A0B3B8A1B404411B8B2920F4A678487	ORTF, 1ère chaîne	Jérome Savary	fr	1	Lot3_Topic11_a_2010_09.zip		Ina_CPF86649487_J-01308.mp4
9336	EUS_AC2079F866B84E5CAED3F7E15A14A291	ORTF	Edith Piaf	fr	1	Lot3_Topic11_a_2010_09.zip		Ina_CPF03007301_J-04746.mp4
9337	EUS_6A9FC318C0B841809474B61990333A45	ORTF	Jacques Brel	fr	1	Lot3_Topic11_a_2010_09.zip		Ina_RAF03002143_J-04764.mp4

Figure 8.3 The first page of the Annotation Services Tool of the Euscreen Ingestion platform.

When the user selects a specific item, it gets highlighted and a number of CRUD (Create, Update, and Delete) functionalities are available, those are the following:

- Delete the item
- View the item
- Add a new item
- Edit an item

By selecting to edit a specific item, a page like the one depicted in (fig. 8.4) is presented with all the metadata fields that are included in the EUscreen Metadata Schema organized in three distinctive sub-forms. Those sub-forms can be hided or un-hided in order to make the whole process of annotating easier. It has to be noted that the annotation process visual representation has a 1-1 correspondence with the EUscreen Metadata Schema and as the user types in data, validation occurs. In this way it is ensured that the items annotated will be complete and valid taking into consideration the EUscreen Metadata Schema. Then, the items are stored in EBUcore schema. The annotation page also contains valuable information like a timeline view of the different versions of the annotated item, if the user wishes at any time he/she is able to view an older version of the item and also compare them. The metadata of the item are groups in the following groups:

- Content Descriptive Metadata (fig. 8.5)
- Item Descriptive Metadata (fig 8.6)
- Administrative Metadata (fig 8.7)

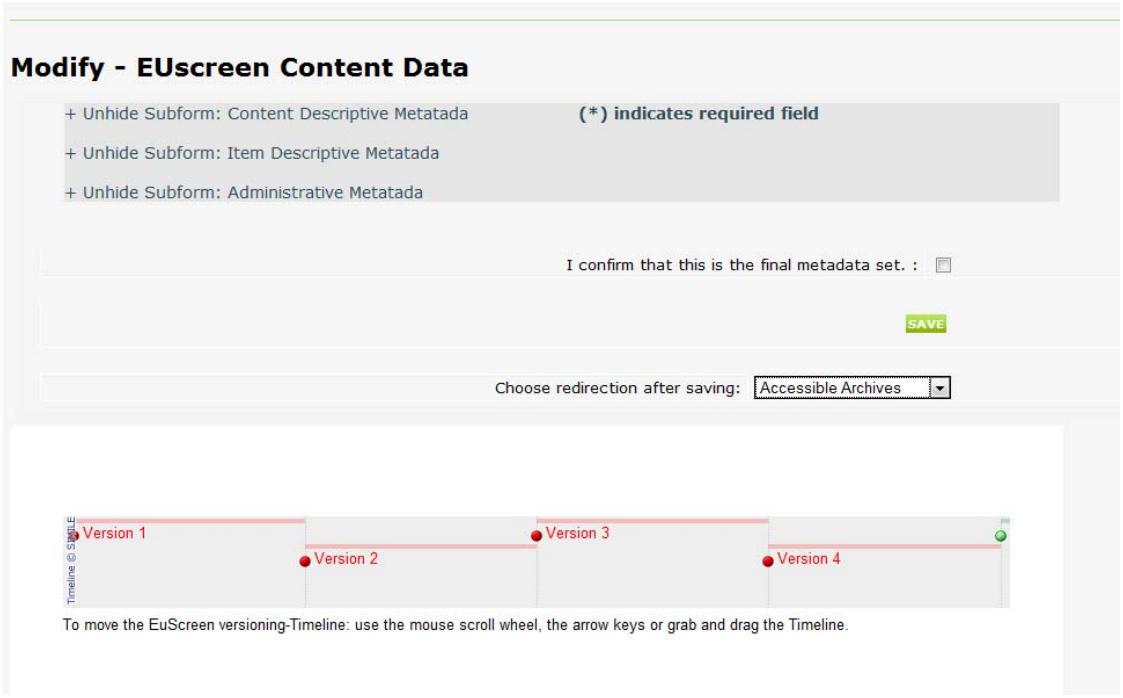


Figure 8.4 The initial page of the annotation process for a specific item, a version timeline is present and the three groups of metadata fields.

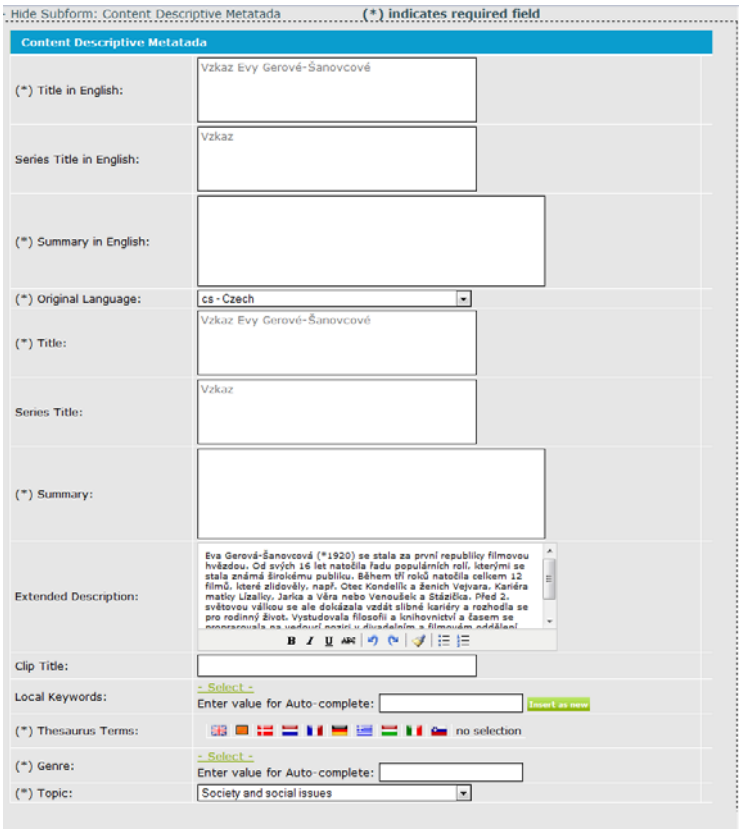


Figure 8.5 The Content Descriptive Metadata sub-set form.

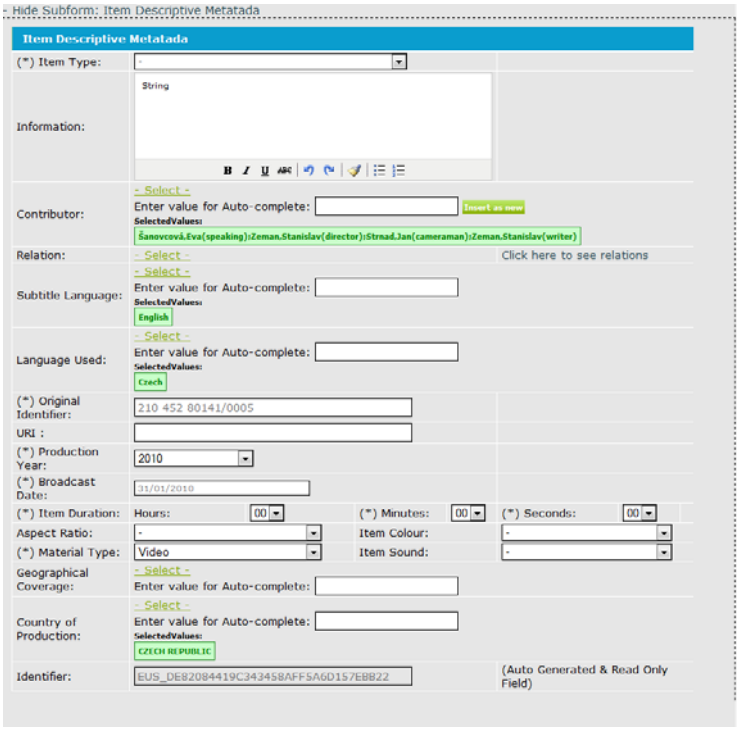


Figure 8.6 The Item Descriptive Metadata sub-set form.

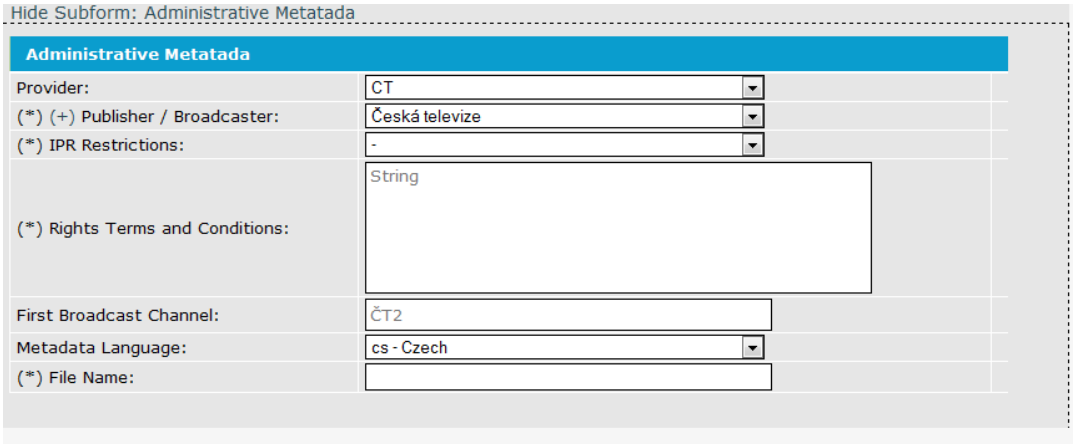


Figure 8.7 The Administrative Metadata sub-set form.

9 Relevant work

This chapter provides an overview of relevant platforms and tools that deal with ingesting, mapping and transforming metadata records as well as with enabling permanent access to digital works.

9.1 The HP-MIT DSpace Repository project

The DSpace project was initiated in July 2000 as part of the HP-MIT alliance. In 2007 the DSpace foundation was formed as a non-profit organization to provide support to the growing community of institutions that use DSpace. The foundation's mission is to lead the collaborative development of open source software to enable permanent access to digital works.

DSpace is a platform that allows you to capture items in any format – in text, video, audio and data in general with the purpose of distributing it over the web. It indexes the data so users can search and retrieve the items that constitute it. Moreover, another major functionality of DSpace is the ability to preserve the data over long term. It is typically used as an institutional repository supporting the following three roles:

- Facilitate **capture** and **ingest** of materials, including any related metadata.
- Facilitate **easy access** to the materials, both by **listing** and **searching**.
- Facilitate the **long term preservation** of the materials.

Finally, DSpace can be used to store any type of digital medium, e.g. videos, images, data sets, journal papers and others. The overall architecture of DSpace is presented in Figure 9.1.

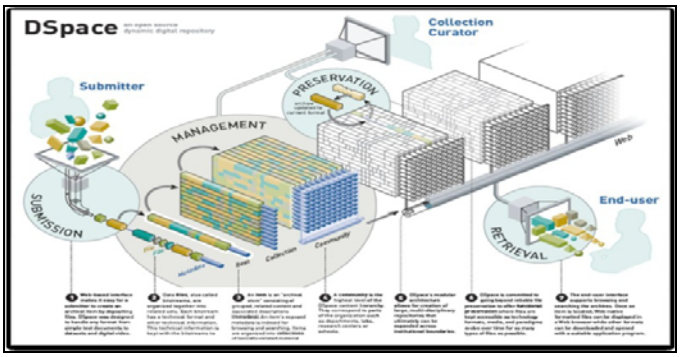


Figure 9.1 the over architecture of the DSpace platform.

DSpace is designed to work "out of the box" for basic repository needs while still being customizable; the system has been put to a wide variety of uses, and has been entrusted with important intellectual content produced by many institutions. While a certain amount of evolution can take place simply by patches, contributions and reimplementation of specific components the DSpace foundation recognized the necessity of a major review of the core architecture of DSpace, motivated mainly from the widespread adoption of the platform and the technological developments that occurred from the initial design of DSpace back in 2000.

For this reason a group of experts and DSpace committers was formed in 2004 that would be responsible of re-evaluating the DSpace platform and propose a set of technical characteristics the DSpace Version 2.0 should have. The outcome of this group can be summarized in the following list of principles. These principles will govern the development of the next version of DSpace.

1. DSpace should be primarily open source software for building digital repositories.
2. DSpace should be usable based purely on free and open source software.
3. DSpace should have a decoupled, stable and application neutral core.
4. DSpace should be usable for a variety of applications but at the same time it will retain useful “out of the box” functionality for common use cases.
5. DSpace should employ and support existing, open standards where possible and practical.
6. DSpace releases should be minimal disruptive.
7. DSpace should support an exit strategy for content.
8. DSpace should evolve.

Based on these design principles the group of experts compiled a list of specific recommendations that would be part of the new version of DSpace. These recommendations attempt to tackle existing issues that appear in the current version of DSpace, e.g. scalability and interoperability among others.

9.2 The Fedora Digital Object Repository Management System

The Fedora digital object repository management system is based on the Flexible Extensible Digital Object and Repository Architecture (FEDORA). The system is designed to be a foundation architecture upon which full featured institutional repositories and other interoperable web based digital libraries can be built. It was jointly developed by the University of Virginia and Cornell University, the system implements the Fedora architecture, adding utilities that facilitate repository management. The current version of the software provides a repository that can handle one million objects efficiently. Subsequent versions of the software will add functionality important for institutional repository implementations, such us policy enforcement, and performance enhancement to support very large repositories. The system’s interface comprises three web based services:

1. A management API that defines an interface for administering the repository, including operations necessary for clients to create and maintain digital objects;
2. An access API that facilitates the discovery and dissemination of objects in the repository; and
3. A streamlined version of the access system implemented as an HTTP-enabled web service.

Fedora supports repositories that range in complexity from simple implementations that use the web service’s “out of the box” defaults to highly customized and full featured distributed digital repositories.

Another characteristic of Fedora is that since it is a web service, it does not have a standard front end. Nevertheless, many UI applications have been implemented to front-end Fedora by the open source community that supports it.

One of the main strengths of the Fedora framework is that it demonstrates the best scalability among the most used repositories that exist. At the same time it easily supports the storage of multiple types of digital objects and collections particularly well. Another noticeable strength of the platform is that as a foundation architecture that provides powerful API based interoperability features, Fedora is highly flexible and powerful, and has proven itself with large networked repositories similar to those envisaged with the OARINZ project. With no set user interface, Fedora has true separation between the 'backend' and 'frontend'. Fedora provides good interoperability among different systems, with different options allowing for smart and flexible integration methods. Finally, it is supported by a strong development team and development map.

In a sense, a key strength can also be perceived as a weakness. With no user interface, Fedora cannot offer a full repository service 'out of the box' and therefore provides a conceptual complexity which other systems like DSpace do not. The code base of the Fedora platform is probably the largest among the commonly used repositories while at the same time the Fedora development community can be described as closed. These two weaknesses reduce the adoption of the Fedora platform by the repository community.

9.3 The EPrints repository platform

The EPrints software has probably the largest and most broadly distributed base of the majority of the repository platforms that exist. It was developed at the University of Southampton and the first version of the system was released in late 2000. The project is supported by JISC, as part of the Open Citation Project and by NSF. EPrints worldwide installed base affords an extensive support network for new implementations. The size of the installed base for EPrints suggests that any institution can get it up and running with minimal effort or technical expertise. Moreover, the number of EPrints installations that have augmented the system's baseline capabilities, for example by integrating advanced search, extended metadata and other features, indicates that the system can be readily modified to meet local requirements.

As already mentioned, the EPrints platform is a good candidate as the repository platform of choice for many institutions because it is one of the least complex systems in existence and hence it has a low skill barrier to implement and maintain. At the same time, because it has one of the widest install bases, it goes a long way to ensure its longevity as a fully supported system. Finally, the code base of EPrints is uniform and well documented making it easier to work on for low level customization.

A major weakness of EPrints lies in the data model used which causes some scalability issues, although these could be addressed with some development effort. Also, its method of adding new digital content type can lead to disparate data models and compatibility issues if maintaining multiple systems. Finally, the development team of EPrints denies any external contribution to the code base of EPrints.

9.4 The CERN Document Server Software (CDSware)

The CERN Document Server Software (CDSware) was developed to support the CERN Document Server. The software is maintained and made publicly available by CERN (the

European Organization for Nuclear Research) and supports electronic preprint servers, online library catalogs and other web based document repository systems. CERN uses CDSware to manage over 350 collections of data, comprising over 500,000 bibliographic records and 220,000 full text documents, including preprints, journal articles, books and photographs.
CDSware was designed to accommodate the content submission, quality control, and dissemination requirements of multiple research units. Therefore, the system supports multiple workflow processes and multiple collections within a community. The service also includes customization features, including private and public baskets or folders and personalized email alerts.

CDSware was built to handle very large repositories holding disparate types of materials, including multimedia content catalogs, museum object descriptions and confidential and public sets of documents. Each release is tested live under the rigors of the CERN environment before being publicly released.

The CDSware exhibits the following major weaknesses:

- It has extremely complex installation steps.
- CDSware also does not have a good community around it. The mailing list has had very limited traffic since 2002, which indicates that this project may have sustainability issues going forward.

9.5 DRIVER: Building a sustainable infrastructure of (European) Scientific Repositories

The DRIVER platform which is the outcome of the European funded e-infrastructure project "DRIVER: Building a sustainable infrastructure of (European) Scientific Repositories", does not constitute a repository platform but a framework for creating and managing a network of existing repositories. The main aims and objectives of the Driver platform are the following:

- To organize and build a virtual, European scale network of existing institutional repositories.
- To assess and implement state-of-the-art technology, which manages the physically distributed repositories as one large scale virtual content resource.
- To assess and implement a number of fundamental user services
- To identify, implement and promote a relevant set of standards
- To prepare the future expansion and upgrade of the DR infrastructure across Europe and to ensure the widest possible involvement and exploitation by users.

Version 1.0 of the D-NET Software: Driver network-Evolution-Toolkit is already released under the Apache open source license to the public including the following modules:

- Repository network administration software (such as the Repository Network Manager, Resource Monitoring and others).
- End User services (search, browse, profiling).
- Support service to local repository managers and aggregators (Validation Tool).

The current Driver infrastructure supports three groups of users, A) the repository manager, B) the service provider and C) the researcher, reader, public. Apart from only providing the appropriate technological tools to support the creation and maintenance of a repository network, Driver also defines and supports the concept of the European Community of repository networks. “Community” means that the members agree to some fundamental principles and that the Driver community is wider than the Driver consortium and has no legal restrictions and is open to new members. Some of these fundamental principles are listed below:

1. Make research publications open to the public.
2. Become partner in a repository service network.
3. Follow “guidelines” to make data and services interoperable.
4. Ensure long term access to an institution’s research publications.

9.6 REPOX – A Metadata Space Manager

Repox is a framework to manage metadata spaces. It comprises several channels to import metadata from data providers, services to transform metadata between different schemas according to user’s specified rules, and services to expose the results to the exterior. This tailored version of Repox aims to provide to the TEL partners a simple solution to import, convert and expose their bibliographic data via OAI-PMH, by the following means:

- Cross platform. Repox is developed in Java so it can be deployed in any operating system that has an available Java Virtual Machine.
- Easy deployment. Repox is available with an easy installer, which includes all the required software and libraries.
- Support for several metadata formats. Repox currently supports MARC21, UNIMARC, MarcXchange and MARCXML schemas out of the box and encodings in ISO 2709 (including several variants).
- Metadata crosswalks. It offers crosswalks for converting MARC21 and UNIMARC records to simple Dublin core as also to TEL-AP. A simple user interface makes it possible to customize these crosswalks and create new ones for other formats.

Repox is not a complete repository platform, although it imports metadata and stores them in a custom format for easy access providing at the same time a way of exposing these metadata to the web using an implementation of the OAI-PMH protocol for exchanging metadata over the web. It also includes a mapping tool capable of mapping various input metadata schemas to the TEL format. For this reason, in its current state Repox is limited to support only the exposure of metadata transformed in the format defined and supported by the TEL project.

10 Conclusion

The deliverable presented the web services that have been implemented in EUscreen project in order to ingest content providers' metadata in EUscreen and Europeana portals. The web services that consist of the EUscreen ingestion tool are the following, the harvesting-delivery, mapping, statistics, transformation and annotation. The relevant work, past and ongoing, has also been presented in this deliverable.