Biblio-transformation-engine: An open source framework and use cases in the digital libraries domain

Kostas Stamatis, Nikolaos Konstantinou, Anastasia Manta, Christina Paschou, Nikos Houssos

National Documentation Centre / National Hellenic Research Foundation
{kstamatis, nkons, amanta, cpaschou, nhoussos}@ekt.gr

1. Introduction

In the course of developing digital libraries, repositories and archives, a constantly recurring requirement is the transformation of data between diverse formats in order to satisfy various needs, which may arise both in-house as well as be raised by end user groups, or simply by technological evolutions. Additionally, the data itself is more important than the code that handles it, and therefore the code changes far more frequently than the data. This highlights the necessity for reusable software that focuses on data transformations.

Stemming from this observation, the hereby presented approach aims at facilitating the often encountered transformation tasks. The source data can be in the form of records in a legacy database, conforming to deprecated formats, or simply satisfying internal ad hoc needs. The target of the transformation is more usually a step towards opening the data, enabling data integration with other sources, by conforming to widely adopted standards and practices. An important observation that provided the motivation for our work is that the overall data transformation task consists of components that can be engineered in a modular way to be largely independent of each other and to achieve a high degree of reuse of some of them, even in very different real-life cases. Therefore, we have created the biblio transformation engine, an open source framework that enables fine-grained modularity and injection of functional elements in a way that allows heavy reuse and thus faster development of transformation tasks. The document is structured as follows: Section 2 provides details on the biblio transformation engine, the tool developed and used in the transformation process in order to alleviate the required corresponding effort. Section 3 illustrates several use cases in real-world applications, while Section 4 concludes the paper by gathering our most important observations and future plans.

2. Biblio transformation engine

The biblio transformation engine is a generic framework for implementing data transformation workflows. It can be used out-of-the box for common transformation cases, as it comes with a wide range of tools specific to particular formats and standards, but is inherently highly configurable and extensible to accommodate any particular transformation needs. It allows the decoupling of communication with third party data sources and sinks (e.g. loading and exporting/exposing data) with the actual tasks that comprise the transformation. Furthermore, it enables the decomposition of a workflow into autonomous, modular pieces (transformation steps), facilitating the continuous evolution/re-definition of workflows to constantly changing data sources and the development of fine-grained workflow extensions in a systematic way.

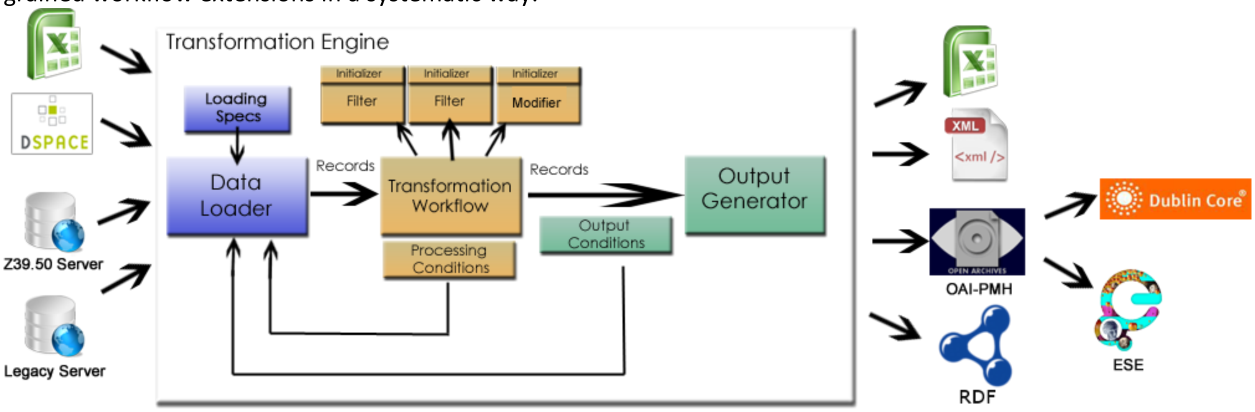

Figure 1. Architecture of the transformation engine.

A key aspect of the engine's design is the Record abstraction with a simple common interface for all types of records that allows complex transformation functions. Examples of record implementations that have been implemented and used until now include UNIMARC, MARC21, Dublin Core, ESE, various structured formats for references (e.g. BibTex, RIS, Endnote), etc. This abstraction enables, for example, software developers to process MARC records without knowledge of the specifics of MARC.

The architecture of the engine is depicted in Figure 1: *Data loaders* are used to read data from external sources (e.g. files, repository databases, Z39.50 servers, even OAI-PMH data providers). The *Output Generators* undertake the exporting, exposing or submission of records to third party systems and applications. Before the generation of output, a processing pipeline is executed consisting of processing steps categorized as *Filters*, *Modifiers* and *Initializers*. *Filters* determine whether an input record will make it to the output. *Modifiers* can perform operations on record fields and their values (e.g. add/remove/update field). *Initializers* initialize data structures that are used by processing steps.

A specific transformation instance is defined as a series of processing steps in a configuration file outside the source code of the engine, in particular using the dependency injection mechanisms of the Spring framework (www.springsource.org). Thus, a transformation engine system can include many data loaders, output generators and transformation steps, but a specific scenario can make use of only some of them according to the user needs. An individual transformation case can be addressed just by configuring the transformation steps in the Spring XML configuration file or, in more complex cases, by developing and injecting into the system any required data loaders, output generators, or processing steps that might not be available.

The biblio transformation engine is available as free software (provided with the EUPL license) at Google Code (http://code.google.com/p/biblio-transformation-engine/). It has been utilized for a variety of use cases as described in the next section.

3. Use cases

This section provides more details about use-cases where the tool as described above finds application.

3.1 Linked Open Data

The semantic web ecosystem nowadays finds application in many cases out of the academia. Technology maturity in the domain has enabled publishing and exploiting semantically annotated datasets, leading to the creation of a wider movement towards *Linked Open Data*, a term that has turned out into a buzzword itself [1]. Linked Data means publishing structured data, in an open format, and making it available for everyone to use it. Using RDF syntax for this is ideal because the data can be interlinked, creating a large pool of data, offering the ability to search, combine and exploit the knowledge. Users can even navigate between different data sources, following RDF links, and browse a potentially endless Web of connected data sources [2].

Technologically, the core idea of Linked Data is to use HTTP URIs to identify arbitrary real-world entities. Data about these entities is represented using the Resource Description Framework (RDF). The RDF model encodes data in the form of subject, predicate, object triples. The subject and the object of a triple are both URIs that identify a resource, or a URI and a string literal respectively. The predicate specifies how the subject and object are related, and is also represented by a URI.

By publishing data on the Web according to the Linked Data principles, data providers add their data to a global data space, which allows data to be discovered and used by various applications. Publishing a dataset in this way involves a couple of basic principles, including:

- Assigning HTTP URIs to the entities described by the data set.
- Defining URIs in an HTTP namespace under your control, in order to be able to make them dereferenceable.
- Including RDF links to other data sources on the web, so that clients can navigate the Web of Data as a whole by following RDF links.

Technically, the means used to achieve semantic annotation relies on the Jena framework (developed by Hewlett Packard and currently an Apache incubator project). An Output Generator was developed, consuming the Data Loader inputs in order to generate the respective RDF description for each of the source records.

3.1.1 The Helios repository

A linked data compliant digital repository enables content re-use, allows participation of individual collections to the evolving global and gives the user the chance to discover new data sources at runtime by following data-level links, and thus deliver more complete answers as new data sources appear on the Web [3]. The Helios repository (http://helios-eie.ekt.gr) is the institutional repository of the National Hellenic Research Foundation (NHRF). Type of items included are publications in international journals, (a limited number of) research data sets, conference articles and proceedings, books, studies, videos from lectures and events.

To achieve a semantically richer representation of the Helios repository we modified the biblio transformation engine so that it loads the data via an OAI-PMH data loader (in a way somewhat similar to [6]) and then turns the Helios dataset to a Linked Data compliant dataset. URIs have been assigned to the following metadata fields: title, creator, contributor, date, language, type, source, pubisher, subject category(LCC), ISSN, coverage spatial. When exposing a new collection as Linked Data, it is a good practise to reuse existing vocabularies/ontologies for its description as this makes it easier for the outside world to integrate the new data with already existing datasets and services [4]. Therefore, existing vocabularies were used for fields like, spatial coverage (http://www.geonames.org/) and ISSN (http://periodicals.dataincubator.org).

3.1.2 The Claros project

The proposed system has been utilised for the delivery of Claros-compliant data from Cycladic museum records (http://www.cycladic.gr) to Claros. Claros [5] is "an international interdisciplinary research federation using the latest developments in Information and Communication Technologies to bring the art of the world to everyone", led by the University of Oxford. It combines information from major research databases in archaeology, which have been developed independently over many years. A variant of CIDOC-CRM (http://www.clarosnet.org/wiki/index.php?title=Main_Page) is used as the basis of a common terminology and ontology for bringing together such diverse information. The dataset can be explored at http://explore.clarosnet.org, queried using SPARQL at http://data.clarosnet.org/sparql/ and therefore offering a valuable resource for academics, professionals, as well as for individuals with interest in art.

The Cycladic museum collection contains approximately 350 objects of Cycladic, Ancient Greek and Cypriot art presented online. Cycladic archives are being maintained and made accessible on-line in two languages, English and Greek. The data is kept within a well-known proprietary museum management system, which provides an export of the items metadata in a custom XML format.

The Biblio Transformation Engine enabled loading of data by utilizing a XML data loader and a RDFOutputGenerator based on the Jena framework. The record mappings followed typical practices for CIDOC-CRM [7] and Claros. Example components that had to be implemented was a date modifier that normalizes dates according to Claros specifications (e.g. define time ranges that enables timeline browsing in the Claros visualisation interface), and a Provenance modifier that removes prepositions and other strings from place names. The modifier logic is totally unaware of XML and the Claros format, which is based on CIDOC-CRM and RDF.

A URI has been assigned to a number of metadata fields: title, id, material, date, provenance, type, thumbnail, culture. Some of them contain links pointing at other vocabularies thereby defining mappings between related vocabularies: places to the geonames database (http://www.geonames.org), types of objects and materials to the respective Claros-defined lists of values.

3.2 Transform bibliography records to repository formats

Bibliographic records of publications in structured formats, for example BibTeX and RIS, are increasingly being utilised by end users in a variety of applications. These include, among others, word processors or typesetting systems (e.g. MS Word, Libre Office, LaTeX) and reference managers (e.g. EndNote, RefWorks, Citeulike, Connotea, Mendeley, Qiqqa, etc.); their combination enables authors of scientific papers to manage in an efficient way their literature reading lists and the citations they use in their articles, for example through automatic transformations between references styles and ability to reuse publications lists and share them among colleagues and groups. This is facilitated by the export support that exists in popular online bibliographic databases (ISI Web of Knowledge, Scopus, Google Scholar and many others) for the aforementioned formats.

Repositories need to have import/export capabilities for these formats to enable both their batch population with bibliographic records from other sources and assist users in reusing metadata from repositories in tasks when compiling citations lists in their papers and organizing their bibliographies. Using the biblio transformation engine we have implemented the following two use cases for the DSpace platform: (a) Batch import of BibTeX, RIS and ISI Web of Knowledge records into DSpace (i.e. a particular step in a DSpace workflow), (b) Batch export of DSpace records, available through a button in the user interface (in search and browse results) to a variety of formats (e.g. BibTeX, RIS, text) and reference styles. For (a) we needed to create data loaders for the respective formats and reused the configurable output generator producing records in the DSpace import structure; we had already developed it for migrating data into DSpace from other types of records (e.g. Excel, MARC). In specific cases, we had also to create modifiers, for example for expanding abbreviated journal names. For (b) we reused an existing data loader for a DSpace database (initially developed for the DRIVER/OpenAIRE use cases described in 3.3) and developed a generic, configurable output generator that uses REST web services over the citeproc-js citation processor to produce records in the desired formats. The citeproc-js module is able to transform anything expressed in the Citation Style Language (CSL) and in our proposed configuration runs in a Node.js server.

A similar category of use cases concerns migration of data from bibliographic catalogs to repositories. The biblio transformation engine has been used for this purpose in a variety of cases, the most complex of whom was the Hellenic National Archive of Doctoral Dissertations [8], where an initial loading of more than 24.000 UNIMARC records of PhD theses to a DSpace repository took place, involving a UNIMARC data loader and many other components (mainly modifiers) for quite complex data cleaning and mapping tasks.

3.3 Feeding content aggregators

A number of important aggregators with international coverage and diverse scope have entered the scene in the last few years. Distinctive examples are Europeana, the European digital heritage gateway, DRIVER and OpenAIRE (repositories of peer-reviewed scientific publications) and DART Europe (European portal to research theses). This paragraph describes how the biblio transformation engine has been used to feed all four of these aggregators.

Compatibility with aggregators is nowadays a pre-requisite for repositories. In this context, it is becoming an increasingly common requirement for repositories to provide for retrieval by an aggregator only a subset of the metadata records it contains that meet specific criteria, essentially enabling selective harvesting. Moreover, some aggregators collect records for content in specific subject areas, while individual repositories can be interdisciplinary. Such is the case with the VOA3R aggregator on Agriculture and Aquaculture and Europeana since is concentrates on collecting mainly cultural heritage content. At last, some aggregators collect only records for content of a specific type (e.g. theses, like DART Europe), while individual repositories may contain different types.

The above indicate the complexity of supporting selective harvesting. This requirement becomes more difficult to achieve when you consider that a repository is likely to provide records to more than one aggregators, each with different requirements. Another important aspect and use case of selective harvesting is the retrieval of records from systems that are not compliant with OAI-PMH. These might include legacy systems like custom, databases and bibliographic catalogs of Integrated Library Systems connected with the corresponding digital material, etc.

All of the above cases and the aforementioned issues were addressed successfully with the use of the biblio-transformation engine. For example, in the challenging case of providing to European from legacy systems, this was achieved by incorporating the biblio-transformation engine within an OAI-PMH server implementation (OAICat OCLC) and thus, enhancing the OAI-PMH server to get data from various sources and perform selective harvesting, while respecting the OAI-PMH “contract” towards clients [9].

For the three remaining cases, in terms of implementation of, a single data loader was implemented to retrieve data from the DSpace database and was reused for all of them. Selective harvesting required the development of specific filters - certain of them were reused (e.g. the “digital file available” filter). Different output generators were implemented for each case. Due to configurability of the engine, based on an external XML file, the same engine instance was used to produce totally different results depending on the needs of a particular aggregator.

4. Conclusions and lessons learnt

The current project cannot aim at being a one-solution-fits-all, Swiss Army knife of the typically painful and tedious job of data transformation, but it provides an open source framework that may work out-of-the-box with reasonable configuration effort for common cases in the areas of digital libraries and repositories and can be easily extended in a piece-wise manner to accommodate more complex cases. Our approach, already with many applications in real-life situations and in different environments in terms of technology, standards and functional requirements, has shown that modularity and separation of concerns (e.g. a developer modifying MARC records without having expertise in MARC) is perfectly possible in that context, leading to considerable code reuse and efficient development of transformation functions. Experience until now shows that data loaders and output generators are the components that are more heavily reused, followed by filters. Modifiers are the category of functional elements that are more difficult to generalize and reuse.

Our future plans include support for more formats to import/export, providing more configurable data processing workflows and more general and formal ways to specify and implement mappings among data sources. At all times, contributions from the community are welcomed and encouraged, especially for components (data loaders, output generators and filters/modifiers) that could be applicable to common cases of transformations in the international digital libraries and repositories community.

References

[1] Bizer, Christian and Heath, Tom and Idehen, Kingsley and Berners-Lee, Tim: Linked data on the web (LDOW2008), Proceedings of the 17th international conference on World Wide Web, Beijing, China, 2008.

[2] C. Bizer, T. Heath, T. Berners-Lee: Linked Data - The Story So Far. Int. J. Semantic Web Inf. Syst. 5(3): 1-22 (2009)

[3] Stevenson A.: Linked Data - The Future of Open Repositories?, Open Repositories 2011.

[4] Knoth, P., Robotka, V. , Zdrahal, Z.: Connecting Repositories in the Open Access Domain using Text Mining and Semantic Data, International Conference on Theory and Practice of Digital Libraries 2011 (TPDL 2011), Berlin, Germany.

[5] Kurtz D, Parker G, Shotton D, Klyne G, Schroff F, Zisserman A, Wilks Y: CLAROS-bringing classical art to to a global public, Proc. IEEE e-Science Coference, Oxford, 9-11 December 2009.

[6] B. Haslhofer and B. Schandl, “The OAI2LOD Server: Exposing OAI-PMH Metadata as Linked Data,” in Proceedings of WWW 2008 Workshop Linked Data on the Web (LDOW2008), , Beijing, China, 2008.

[7] M. Doerr: Mapping of the Dublin Core Metadata Element Set to the CIDOC CRM. Technical Report 274, Institute of Computer Science, Foundation of Research and Technology, Greece, 2000.

[8] N. Houssos, P. Stathopoulos, I. Sarantopoulou, D. Zavaliadis, E. Sachini, Evi (2010) A service-oriented national e-theses information system and repository. Open Repositories 2010, Duraspace User Group.

[9] N. Houssos, K. Stamatis, V. Banos, S. Kapidakis, E. Garoufallou, and A. Koulouris: Implementing enhanced OAI-PMH requirements for Europeana, In Proceedings of the 15th international conference on Theory and practice of digital libraries: research and advanced technology for digital libraries (TPDL'11), Springer, 2011, pp. 396-407.