Library Review

Enhanced OAI-PMH services for metadata sharing in heterogeneous environments

Nikos Houssos Kostas Stamatis Panagiotis Koutsourakis Sarantos Kapidakis Emmanouel Garoufallou Alexandros Koulouris

Enhanced OAI-PMH services for metadata sharing in heterogeneous environments

Received 7 May 2014
Revised 11 June 2014
Accepted 27 June 2014

Nikos Houssos, Kostas Stamatis and Panagiotis Koutsourakis
National Documentation Centre, National Hellenic Research Foundation, Athens, Greece

Sarantos Kapidakis
Department of Archive, Library and Museum Sciences, Ionian University, Corfu, Greece

Emmanouel Garoufallou
Department of Library Science and Information Systems, Alexander Technological Educational Institute of Thessaloniki, Thessaloniki, Greece, and

Alexandros Koulouris
Department of Library Science and Information Systems, Technological Educational Institute of Athens, Athens, Greece

Abstract

Purpose – This paper aims to propose a toolset that enables individual digital collections owners to satisfy the requirements of aggregators even in cases where their IT and software infrastructure is limited and does not support them inherently. Managers of repositories/digital collections face the challenge of exposing their data via Open Archives Initiative – Protocol for Metadata Harvesting (OAI-PMH) to multiple aggregators and conforming to their possibly differing requirements, for example on output metadata schemas and selective harvesting.

Design/methodology/approach – The authors developed a software server that is able to wrap existing systems or even metadata records in plain files as OAI-PMH sources. They analysed the functionality of OAI-PMH data providers in a flow of discrete steps and used a software library to modularise the software for these steps so that the whole process can be easily customised to the needs of each pair of OAI-PMH data provider and service provider. The developed server includes a mechanism for the implementation of schema mappings using an XML specification that can be defined by non-IT personnel, for example metadata experts. The server has been applied in various real-life use cases, in particular for providing content to Europeana.

The work of the three authors from the National Documentation Centre/National Hellenic Research Foundation has been partly supported by the projects "National Information System for Research and Technology – Social Networks and User Generation Content" (Ref No 296115) and "Platform for provision of services for deposit, management and dissemination of Open Public Data and Digital Content" (Ref No 327378), which are co-funded by the Greece and the European Union-European Regional Development Fund through the Operational Programme "Digital Convergence" (NSFR).

Library Review
Vol. 63 No. 6/7, 2014
pp. 465-489

0024-2535
DOI 10.1108/LR-05-2014-0051

Findings – It has been concluded through real-life use cases that it is indeed possible and feasible in practice to expose metadata records of digital collections via OAI-PMH even when the data sources do not support the required protocols and standards. Even advanced OAI-PMH features like selective harvesting can be supported. Mappings between input and output schemas in many practical cases can be implemented entirely or to a large extent as XML specifications by metadata experts instead of software developers.

Practical implications – Exposing data via OAI-PMH to aggregators like Europeana is made feasible/easier for digital collections owners, even when their software infrastructure does not inherently support the required protocols and standards.

Originality/value – The approach is original and applicable in practice to diverse technology environments, effectively addressing the indisputable fact of the heterogeneity of software and systems used to implement digital repositories and collections worldwide.

Keywords Interoperability, Institutional repositories, OAI-PMH, Metadata sharing, Selective harvesting, Information integration, Europeana, Europeana semantic elements, Biblio transformation engine

Paper type Research paper

1. Introduction

The proliferation of repositories worldwide has created a favourable environment for the emergence of content aggregators that collect metadata-only records from individual data sources and at a minimum provide unified search and browse functionality. Sharing metadata to third parties, including aggregators, has become one of the major functions of scientific and cultural repositories and at the same time a challenge for their managers and developers.

Repositories and digital libraries – applications that facilitate the management of library, archive and museum content in a digital form (Giannakopoulos *et al.*, 2012) – are widely distributed among European countries. A broad range of standards including various formats, different content types and multiple metadata schemes are used. This information, either in the cultural or in the scientific sectors, should be accessible to European citizens, through a platform for sharing and disseminating knowledge (Garoufallou and Asderi, 2010; Garoufallou *et al.*, 2010). To cope with this need, various aggregation schemes have emerged. Europeana (the Digital Library, Archive and Museum of Europe) is an evolving service that tries to be a single access point for Europe's cultural heritage. According to IRN Research (2011), Europeana is of vital importance for European cultural awareness. The Europeana service (Koninklijke Bibliotheek, 2009) is designed to increase access to digital content across Europe's cultural organisations (i.e. libraries, museums, archives and audio/visual archives). Thus, it will constitute an umbrella of European metadata from distributed cultural organisations. Europeana currently provides access to more than 26 million items representing all types of materials including films, photos, paintings, sounds, maps, manuscripts, books, newspapers and archival papers. This process brings together and links up heterogeneously sourced content, which is complementary in terms of themes, location and time. In February 2014, Europeana's active partner network consists of 2,200 organisations from 33 countries. This network builds on the importance of local identity, multiculturalism and multilinguality in Europe that is achieved via a multilingual digital library like Europeana (Vassilakaki and Garoufallou, 2013).

To achieve these goals the European Union (EU) launched various projects. One of the most fruitful was EuropeanaLocal (2008), which ran from 1 June 2008 to 31 May 2011. This project was designed to involve and support local and regional libraries, museums, archives and audio-visual archives to:

- make the enormous amount of content that they hold available through Europeana; and
- deliver new services.

The project was funded under the eContentPlus Programme of the European Commission. It resulted in a Best Practice Network of distributed and interoperable repositories. EuropeanaLocal had 32 partners from 27 countries, 1,031 plus person months and a €4.3 million budget. By February 2014, EuropeanaLocal partners had made available to the Europeana live service almost six million items. Over 800 organisations that provided content mobilised across 27 countries, and enabled and motivated local institutions and their staff to participate in Europeana by enhancing the skills and expertise of key staff involved in the project.

EuropeanaLocal also had a great impact on Europeana strategy and awareness, documentation and guidelines, workflows and tools and support. For example, EuropeanaLocal promoted aggregation, provided information systems and standards in use, helped in improving the Europeana Semantic Elements (ESE) scheme, evolved the Europeana Data Agreements and first tested and provided feedback in tools like the ESE XML Schema validations. Additionally, EuropeanaLocal partners benefited by learning how to install OAI-PMH[1] repositories, better understanding the importance of metadata and its impact on search results, networking themselves and tuning harvesting procedures (Rowlatt *et al.*, 2011). Part of the content that was provided to Europeana via the EuropeanaLocal project came from Greek cultural organisations that built interoperable repositories as a result of participating in this state-of-the-art network.

In conclusion, even today (March 2014), two years after the end of the EuropeanaLocal project, the network has contributed 26 per cent of the total of Europeana content. Technical and interoperability challenges were overcome, the network has made tremendous progress in content aggregation and the European aggregators' infrastructure was enhanced. However, long-term systemic problems such as financial problems and availability of qualified staff remained (Rowlatt *et al.*, 2011). It is worth noting that currently more content is delivered to the Europeana service not only from completed projects like EuropeanaLocal but also from ongoing projects and initiatives.

In Greece, the diversity of content, skills, repositories and infrastructure implies practical aggregation problems. Academic libraries and the Hellenic National Documentation Centre (EKT) aggregate their repositories' content through the openarchives.gr service, which is the Greek digital libraries search engine. The engine was developed by one of the authors of this paper (Banos) as a freelance service and is now maintained by EKT. The openarchives.gr has (as at May 2014) more than 450,000 records from 68 repositories, using mainly simple and qualified Dublin Core (DC). Greek cultural organisations did not make any aggregation progress until the EuropeanaLocal project (2008-2011). The usefulness of the software tools described below, such as the "Hellenic Aggregator" implemented in 2010 by the Veria Central Public Library (2010), the “Open Archives Engine” (Banos, 2009) and the “oai.pmh validator” (Banos, 2011), helped the content providers to build interoperable repositories and allowed them to provide their content in Europeana.

This support was both in technical issues and in tackling metadata compatibility problems. For example, most repositories that participated in the openarchives.gr search engine have implemented simple DC. The ESE schema (The Europeana Office, 2012) needs more elements like the type of content, divided into text, image and video, and other specific data elements. The content providers were helped technically by the Greek EuropeanaLocal team in batch importing of the ESE metadata fields and values. For example, the “DSpace plugin for ESE” (Banos, 2010), developed by the Veria Central Public Library (VCPL) and EKT (Houssos *et al.*, 2011), was a useful tool for batch importing.

If we take into account that, according to openarchives.gr, the digital content in Greece that is provided mostly by the academic and research sector amounts to 430,443 records; the 136,223 records that the Hellenic Aggregator provides to Europeana is of great significance. About one-fifth of the Greek digital content and almost all the cultural heritage Greek digital content is harvested by the Hellenic Aggregator (Garoufallou *et al.*, 2013).

The first step in the process is to use the Europeana XML Namespace http://europeana.eu/schemas/ese/ and augment existing systems’ configuration to support the additional ESE elements. After implementing ESE support, the repository has to be populated with the appropriate metadata values. This task can be either performed manually through the appropriate user interface of each digital library or automatically by using special software tools developed for this purpose.

Although modern digital repository platforms are increasingly being used, there are also numerous digital libraries built with older or closed source technologies or legacy software which do not support OAI-PMH or any other form of automatic metadata exchange. In these cases, special techniques can be applied to extract metadata through plain HTTP requests, for example the DEiXTo tool (Ntonas and Kokkoras, 2007) for extracting structured metadata out of plain web pages. However, a mechanism is required to provide the extracted data to clients (e.g. aggregators like Europeana) through the OAI-PMH protocol. Furthermore, there are some efforts to provide metadata records from research/academic repositories as linked open data (Konstantinou *et al.*, 2014).

This paper builds on previous work on enhanced OAI-PMH services for Europeana (Houssos *et al.*, 2011). It analyses a toolset for owners of data collections that need to provide metadata records to third parties and in particular Europeana. The toolset aims to assist with common challenges in this process such as non-OAI-PMH-compliant legacy systems, difficulties in implementing schema mappings and lack of support for sophisticated selective harvesting. Focus is placed on the ability to address a variety of cases regarding the maturity level of existing infrastructures for data providers and thus the applicability of the proposed solution to heterogeneous environments.

The structure of the rest of the present text is as follows: Section 2 describes the advanced harvesting requirements addressed by our solution and the motivation based on practical needs of data providers. Section 3 presents related work and Section 4 elaborates on the actual solution. Section 5 describes the application of the proposed approach in real use cases, while Section 6 of the article provides summary, conclusions and plans for further work.

2. The case for enhanced OAI-PMH-compliant data providers

The ubiquitous OAI-PMH protocol provides an interoperability framework based on metadata harvesting. Two types of entities exist in a typical OAI-PMH interaction: the data provider that exposes metadata to interested clients and the service provider that offers value-added services on top of metadata collected from data providers.

A major category of OAI-PMH service providers are aggregators, providing unified search and browse functionality as well as the foundation and infrastructure for advanced value-added services that become particularly meaningful when provided over content of substantial size. A number of important aggregators with international coverage and diverse scope have entered the scene in the past 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 and dissertations).

Compatibility with aggregators is nowadays a sine qua non pre-requisite for repositories, as it provides increased visibility, enables content re-use and allows participation of individual collections in the evolving global ecosystem of interoperable digital libraries. 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, essentially enabling selective harvesting. This may be needed for various reasons; certain indicative use cases include the following:

- The aggregator collects only records that meet specific criteria concerning IPR, copyright and open access:
  - Records are included in the harvesting set only when there is a freely accessible digital item (e.g. full text articles, books, etc.). Such policies are followed by Europeana, DRIVER, OpenAIRE and DART-Europe.
  - Only metadata records which are themselves freely available for various uses, ideally through appropriate licensing (e.g. Creative Commons). This is required, for example, by Europeana.
- Thematic aggregators collect only records for content in specific subject areas, while individual repositories can be interdisciplinary. Such is the case with the VOA3R[2] aggregator on Agriculture and Aquaculture. Europeana can also be considered an analogous example, as in initial stages of development, it concentrates on collecting mainly cultural heritage content (e.g. peer-reviewed journal articles are not included).
- The aggregator collects 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 aggregator, each with different requirements. Typically, OAI-PMH sets are implemented within repository platforms in a static fashion, through the creation of one set per individual collection in the repository. This approach is clearly not sufficient because, as is evident from the above examples, the desired sets to harvest may contain records spread over different collections. For practical needs to be satisfied and capabilities provided by the OAI-PMH sets specifications to be fully exploited, more sophisticated mechanisms are required, for example "virtual" sets that are dynamically formed per request based on specific conditions – a solution perfectly compatible with OAI-PMH.

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, non-standard databases, bibliographic catalogues of Integrated Library Systems connected with the corresponding digital material, etc. A common case is that such systems contain an array of diverse records, many of them not relevant for particular aggregators. Therefore, filtering needs to be applied, possibly according to complex criteria with a local, collection-dependent character. Crucial aspects for the success of this task are:

- the adoption of a systematic way of implementing and injecting into the harvesting logic the filtering functionality; and
- the ability for periodic and incremental execution of the harvesting procedure that enables updates of metadata in the aggregator reflecting changes of records within the source systems.

It is worth noting that the optimal option for content providers of this kind would be to provide their digital content through a repository platform, so that a holistic, standards-compliant solution is applied for the management of their digital material and metadata, enabling advanced services such as digital files preservation, curation, persistent identification, full-text indexing, etc.; however, this might not be feasible in the near term (e.g. due to lack of resources).

Addressing the above requirements and issues constitutes the main aim of the system and approach presented in this paper, elaborated in Section 4.

3. Related work

Mazurek *et al.* (2009) present the idea, role and benefits of a selective harvesting extension of the OAI-PMH protocol, developed and applied in Polish digital libraries in frame of the ENRICH project. Specifically, they describe the OAI-PMH protocol extension developed by the Poznan Supercomputing and Networking Center, which allows harvesting of resources based on a search query specified in the Contextual Query Language. This selective harvesting extension is being used by the Polish national aggregator, which enables extended selective harvesting at the national level. It is notable that in this approach, filtering criteria are specified directly from the side of the aggregator.

The concept, implementation and practical application of the OAI-PMH protocol extension were also presented at Mazurek *et al.*'s (2005) JCDL 2009 poster.

Finally, Sanderson *et al.* (2005) briefly contrast the information retrieval protocols SRW/U (the Search/Retrieve Web service) and OAI (Open Archives Initiative), their aims and approaches, and then they describe ways in which these protocols have been or may be usefully co-implemented.

A common limitation of the aforementioned approaches is that data are retrieved from data sources through queries in standard query languages like CQL. In practical situations it is frequently the case that such queries cannot fulfill the custom and complex selective harvesting requirements for data providers, as demonstrated also in the use case of paragraph 1. Furthermore, this solution requires a full-fledged query language to be implemented against a variety of data sources, while the approach proposed in this paper requires data providers to implement only the specific bulk data loaders and filters that are necessary in their particular case.

The University of Minho (2011) has developed an OAI Extended AddOn for DSpace (2011), which enables selective harvesting through the incremental, piece-wise addition of objects like filters in the OAI-PMH server. The solution is bound to DSpace and does not support retrieval from legacy, non-OAI-compliant sources, as, compared with our approach, there is no abstraction either of the data records or the data loading and output generation functionalities.

The preparation of Z39.50 sources for harvesting via the OAI-PMH protocol is addressed by the European Library in the TELplus project (Freire and Reis, 2009) in a thorough manner with many practical examples and considerations. This work, concerning a particular practical aspect of high importance for a specific case of data source but not a general framework, is very useful to take into account in the harvesting of Z39.50 sources with our mechanism.

4. An innovative approach to creating OAI-PMH data providers

The main idea of our approach is to create a complete toolset for data providers that enables them to expose their information via an enhanced OAI-PMH server. This toolset features advanced capabilities related to the specification of mapping between input and output metadata formats, implementation of the required data transformation and selective harvesting. The toolset is designed to work even in cases where the infrastructure of the data provider is very minimal, for example where there is no OAI-PMH-compliant repository and no significant information technology (IT) human resources to implement the full workflow of transformations. These capabilities are the following:

- A simple, declarative way of expressing mappings between input and output formats, covering the most common cases of transformations between metadata schemas. The mappings are specified in XML configuration files, outside the source code of the system, so they can be edited by non-IT personnel (e.g. metadata experts in the library).
- Modularisation of the steps involved in transforming data and providing it through OAI-PMH. The overall workflow is divided into discrete pieces that can be developed independently of each other. Every piece can be reused in various data transformations. Each data transformation is a workflow that can be possibly built out of a set of existing components. If specialised, not-already-available functionality is needed, it can be smoothly added to the workflow as an extension, developed by IT personnel.
- Definition of dynamic, "virtual" OAI-PMH sets, spanning various repository collections (e.g. "records with items licensed under Creative-Commons-Zero", "records of language Greek").

To achieve the above, we have designed according to these principles and developed a modular enhanced OAI-PMH server that has been successfully used in real-life systems for the following use cases:

- introduction of advanced selective harvesting functionality in OAI-PMH-compliant repositories; and
- implementation of OAI-PMH data providers over data sources that do not support OAI-PMH such as Z39.50-compliant bibliographic catalogues and even plain XML exports of metadata records.

A key component of the enhanced OAI-PMH server is an autonomous library called the biblio transformation engine (BTE), which we developed and utilised in this work.

The rest of Section 4 is structured as follows: First, we describe the BTE architecture and the workflows it supports, then we elaborate on features of our solution concerning metadata abstractions and specification and implementation of schema mappings. Finally, a report on two distinct real-life use cases is provided.

4.1 The BTE

Existing data sources are quite heterogeneous, adopting a variety of metadata schemas and syntaxes, and different ways of accessing them. Data may not necessarily be in XML syntax, and may be provided by non-standard APIs. Different steps (including conversions, filtering, enrichment, etc) are needed to convert them to a common format, which is required to store them in the service provider and build services on top of them. Furthermore, it may be appropriate to add specific information, such as constant values, to all records of a collection; these values may be omitted in the explicit metadata because they are obvious to the user of the collection, but they need to be present when the records are part of a larger set of collections. For example, the records for the Parthenon Frieze may not mention Parthenon or Acropolis anywhere, but these terms/labels should be added when mixed with other records.

The BTE[3] (Stamatis *et al.*, 2012) is a programmatic framework for the implementation of data transformation workflows covering these requirements. 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 transformation into a workflow consisting of autonomous, modular pieces (transformation steps). This facilitates the continuous evolution of workflows to constantly changing data sources and the development of fine-grained workflow extensions in a modular way. The engine is written in Java and constitutes an independent component used in a modular fashion in the proposed toolset. It has been utilised by EKT as an autonomous module for dozens of transformations in real systems, for example for populating the digital repositories of Greek public libraries (Sidhunata *et al.*, 2011) with metadata from ILS catalogues. It is also part of the core distribution of the DSpace repository platform since version 3.0[4], utilised as the basis for batch data import functionality.

Each BTE run consists of three distinct steps: data loading, transformation workflow and output generation. Data is modelled as *records*. Using an abstraction for the record, BTE allows the user to read data from a source in a specific format, modify values of specific fields, filter out records that do not meet certain criteria and, finally, produce output in a possibly completely different format. The framework is built around abstractions for each of these concepts and enables reuse of individual pieces across multiple transformations.

The architecture of the BTE is shown in Figure 1. The input and output abstractions are the data loader and the output generator, respectively. The data loader abstracts the process of retrieving the input data from its source and parsing it into BTE records. It is clear that a different data loader is needed for each type of input schema and format. A range of data loaders have already been implemented for BTE and are publicly available for reuse, including commonly used formats such as Dublin Core, MARCXML, BibTeX and CSV, and protocols like Z39.50, OAI-PMH and more. The output of the data loading procedure is a set containing all the records read from the input source.

Similarly, the output generator interface provides methods for exporting records to a specific format; thus, a separate output generator is required per format. A range of output generators are available with BTE, such as Dublin Core, ESE, CSV, Excel, DSpace XML import format and others. Besides formats, several output types are available such as writing to files, directly saving output to a database, instantiating Java objects or producing XML records to be used by another part of the application invoking the BTE (e.g. an OAI-PMH data provider).

The transformation workflow is executed right after data loading and before output generation. It consists of discrete processing steps of the following two types: *Filters* determine whether an input record will make it to the output based on particular conditions (e.g. record type must be "PhD thesis"); *Modifiers* can perform operations on record fields and their values (e.g. add/remove/update field). Typical use cases for modifiers are data normalisation and cleaning tasks on data fields (e.g. normalisation of date values). An important innovation of the BTE is that the transformation steps operate on data fields and are independent of the input metadata schemas. For example, the same DateNormalizationModifier may process date data fields from MARC, Dublin Core or MODS records, after some suitable XML configuration.

A critical component of any transformation is the implementation of the mapping between the input and the output schemas and formats. The BTE provides a range of choices for the incorporation of mapping logic in the system. Mapping can happen within the data loader or the output generator, while a combination of approaches is also possible (e.g. having the mapping implemented partially by the data loader and complemented with certain Modifiers in the transformation workflow). We will elaborate on mapping issues in Section 4.2.

Particularly useful in practical cases is the feature of BTE of incremental retrieval, processing and export of records. Essentially, each of the data loading and the transformation workflow phases can be repeatedly executed before certain conditions

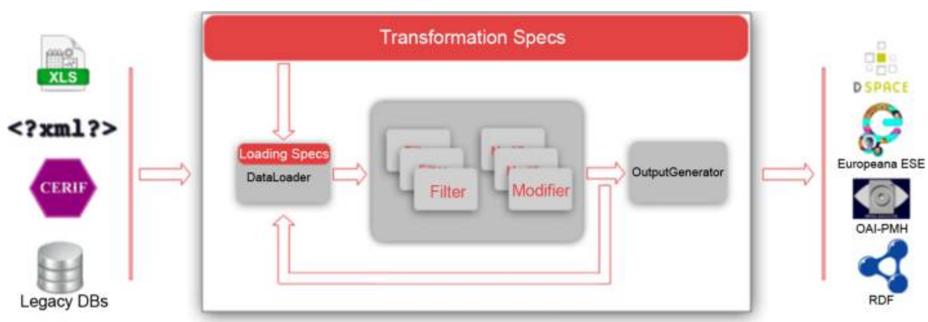

Figure 1.
Biblio transformation engine architecture

are met. This is useful, for example, for enforcing and controlling gradual execution of individual parts of the entire transformation, which might be necessary due to technical limitations introduced by different components, for example as in the following:

- "A maximum of 100 records can be retrieved from the Z39.50 data source per request."
- "A maximum of 10.000 records can be fed to the transformation workflow at a time."
- "Generation of output is meaningful only after a minimum of 1.000 processed records is already available."
- "A maximum of 20.000 records can be forwarded to the output generator at a time."

In summary, implementing transformations with the BTE leads to three major benefits:

(1) Reuse of the same pieces of transformation logic across different transformations. Reuse is specified in a straightforward manner in configuration files editable by metadata experts. Therefore, if certain transformation components are already available as parameterised data loaders, filters, modifiers or output generators in BTE, they can be included to a specific new workflow, saving the effort needed to re-develop them. The contents of the workflow and its parameters, for example schema mappings, can be specified in XML files by non-IT personnel (e.g. metadata librarians).

(2) Separation of concerns is achieved in development. For example, knowledge of the specifics of MARC is not necessary for a developer to create a modifier that performs some changes on an input MARC record.

(3) Metadata records can be incrementally retrieved, processed and exported. This enables execution of workflows in environments where technical limitations apply, for example due to resource-constrained infrastructure or restrictions on the rate/volume of data access by external providers.

4.2 Metadata abstraction and schema mapping in the BTE

As mentioned above, the BTE is built around abstractions for some basic concepts. The central abstraction is that of the *Record*. A simple record is an immutable, read-only object that maps strings, representing field names, to lists of values. The basic functionality provided by the Record public interface is that of retrieving the values corresponding to a given field name, e.g. getByName ("dc.title"). There are no practical limitations for the string used for retrieval, for example it can be an entire XPath expression. Records of any complexity and structure (flat, hierarchical, graphs of entities) can be represented. There is also a mutable (read-write) record abstraction that provides methods for modifying specific values, or even whole fields. We should note that the mutable record is a sub-interface of record, and therefore can be used instead of a record if this is needed. The current implementation of BTE provides two concrete implementations for record: a MapRecord (mutable record) and an XPathRecord (immutable record), but of course software developers using the BTE are free to define their own solutions.

One of the design goals of the framework was to completely decouple the record abstraction from the input and output format of the data. This means that the data loading and the output generation procedures can be parameterised according to the specific needs of each transformation.

The data flow in the BTE, depicted in Figure 2, can be described as follows:

The data loader reads records from their source and transforms them to BTE ***Record*** instances. During the transformation workflow the BTE records are processed and filtered and then they are forwarded to the output generator, which produces the output records in the desired format. The mapping between the input and output schema may be implemented in various places within the BTE, for example the data loader or the output generator. Some, usually advanced, mapping logic may be also incorporated in modifiers of the transformation workflow.

Simple mappings, defining a 1-1 or N-1 correspondence between input and output data fields can be implemented in data loaders and output generator using generic logic. Such mappings can be easily configured outside the application in XML files. Relevant examples are provided in Section 5.

More complex mapping cases can be handled by adding suitable modifiers in the transformation workflow; however, this requires software development by technical personnel.

Often a mapping can be available as an XSLT. In this case, the XSLT can be executed as part of the data loader or the output generator – depending on whether further processing in the transformation workflow is needed and whether it can be performed more effectively or easily to the input or output schema.

5. Enabling OAI-PMH-compliant harvesting of legacy data sources

Large volumes of valuable content are still hosted in systems that are not compliant with OAI-PMH and thus providing them to aggregators like Europeana is a challenging task. Of course, BTE itself is not able to serve OAI-PMH records, but it can be used in a lower level of an OAI-PMH-compliant server to do the transformation of the legacy records to the OAI-PMH ones. In this section, we describe the mechanisms that we have used to enable the OAI harvesting of legacy data sources using the BTE and two use cases that were encountered during our work and have been addressed successfully with the proposed toolset. It is worth noting that this approach makes the harvesting process periodically repeatable even when the underlying data sources are not OAI-PMH-compatible. Selective harvesting is also possible when appropriate (e.g.

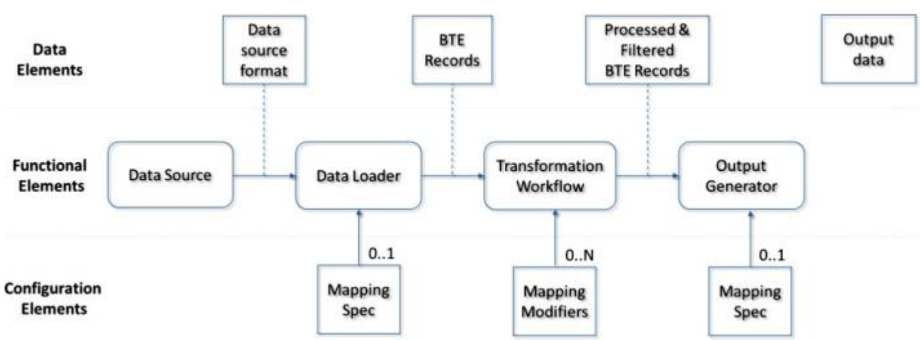

Figure 2.
Data flow in the BTE

harvesting of predefined sites to the source repository is meaningful and/or the source records contain information about their last update date).

5.1 BTE-enabled OAI-PMH server

Our first goal was to create an end-to-end solution for OAI-PMH-based harvesting of legacy data sources. These sources could be a spreadsheet of even a raw XML file or whatever type someone could imagine as long as there is a programming API to load the data in the system. An end-to-end solution means a programmatic interface that could be used by the end-user without caring about the OAI-PMH side of the system but only for the way the records are loaded in the system and how they are mapped to the requested output format/schema. The best way to achieve this was to implement a framework that acts as a middleware layer between the OAI-PMH harvesting and the legacy data sources. The middleware layer we have implemented has the aforementioned features and we are going to describe its internal details in the following paragraphs.

OCLC (Online Computer Library Center) has already implemented a java-based OAI-PMH server (namely, OAI-CAT) with programmatic interface so that the users can extend its functionality -to provide their own functionality for the data loading and the metadata crosswalking. We used OAI-CAT as starting point for our middleware layer. A challenging next step was the procedure to embed the BTE functionality in this workflow to exploit its capabilities of data loading, record filtering and modification and output mapping. Moreover, we needed to expand the configuration options that OAI-CAT provided to us and use the configuration capabilities that BTE offers via its Spring-XML configuration.

The architecture of the implemented middleware solution is shown in Figure 3.

At the very top, OAI-CAT handles the OAI requests. Some of them (i.e. Identify) can be directly resolved by OAI-CAT. However, most of the supported OAI verbs (i.e. ListRecords, ListIdentifiers) cannot be resolved by OAI-CAT itself and thus the proposed middleware and BTE are utilised to feed the OAI responses with appropriate data. Based on the configurations (both OAI-CAT and BTE XML ones), the appropriate data loaders, filters and modifiers are executed and finally the BTE returns to OAI-CAT the corresponding data. The wrapping of the record metadata within the OAI-PMH response is performed by the middleware so the repository owner is only responsible to provide crosswalks between the internal record representation and the requested metadata prefix. The data loading stage can be carried out by the ready-made data loaders (in many flavours) that are bundled with the BTE framework. Otherwise, the repository owner is responsible to provide one and declare it in the configuration. The shaded shapes in the figure correspond to components that might require configuration/extension by the frameworks for the OAI server to work properly.

The OAI-CAT configuration is based on a plain text file named "oaicat.properties". This is mainly to handle the "Identify" response with static data about the data source. The heavy configuration is done in the BTE's Spring-XML configuration file. We mention some of the capabilities of this configuration file and what the user can define within it in the following examples:

- The data loader that will be used to load records (Example 1).

  The user specifies the data loader that will be used to load data in the BTE workflow.

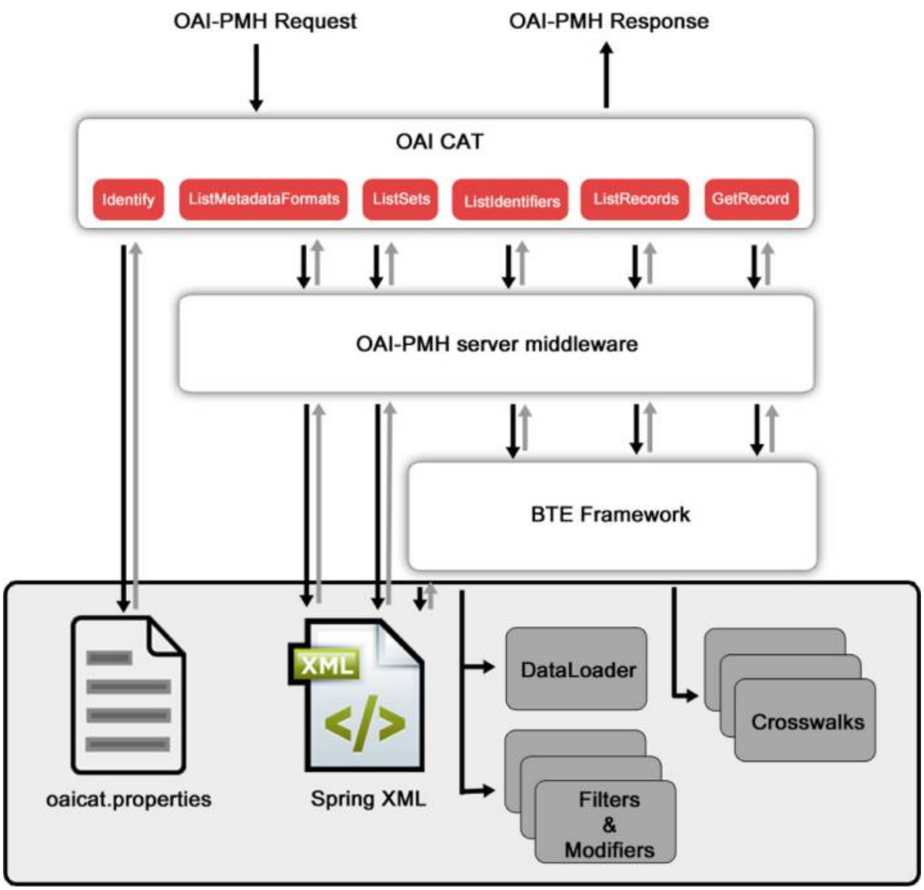

Figure 3. Architecture for OAI-PMH-compliant harvesting of non-OAI-PMH-compliant sources

```
<bean id="dataloader" class="gr.ekt.oai.DSpaceDataLoader" scope="prototype"/>
```

Example 1. Data loader specification

- The BTE transformation workflow that will be used (Example 2).

  This is the default transformation workflow (modifiers) that will be run when records are requested by the OAI server. Keep in mind that the user can specify multiple transformation workflows based on the metadata prefix that will be given in the requested URL. This is very useful when different modifiers need to be applied when harvesters request records in the default oai_dc schema and another metadata schema that this data source supports.
- The metadata formats that the OAI server supports (Example 3).

  Within the first bean, the user declares the list of metadata formats that are supported by this OAI server. The class implementation of the corresponding crosswalks is left to the user, who is the only one that knows the mapping between the records that we loaded by data loader and the requested metadata format by OAI.

- Declaration of virtual sets (Example 4).

  The configuration above declares a new virtual set named "dart" that can be used in the OAI requests. The user is responsible to provide a list of BTE filters that apply when this set is requested, as, in most cases, a defined set actually cuts records from the response.

- The aforementioned middleware can make OAI-PMH harvesting of legacy data sources quite straightforward. This is due to the configuration capabilities provided

```
<bean id="defaultTransformationWorkflow"
                                   class="gr.ekt.bte.core.LinearWorkflow" scope="prototype">
     <property name="process">
          <list>
            <ref bean="fix-metadata-language-modifier"/>
            <ref bean="fix-language-modifier"/>
            <ref bean="field-renamer-modifier"/>
            ...
          </list>
     </property>
</bean>
```

Example 2.
Transformation workflow specification

```
<bean id="crosswalks" class="java.util.HashMap">
   <constructor-arg>
     <map>
       <entry key="oai_dc" value-ref="oaidc-crosswalk"/>
       <entry key="unimarc" value-ref="unimarc-crosswalk"/>
     </map>
   </constructor-arg>
</bean>

<bean id="oaidc-crosswalk"
   class="gr.ekt.oai.OAIDCCrosswalk">
   <constructor-arg>
     <value>
        http://www.openarchives.org/OAI/2.0/oai_dc/
                                        http://www.openarchives.org/OAI/2.0/oai_dc.xsd
     </value>
   </constructor-arg>
</bean>

<bean id="unimarc-crosswalk"
   class="gr.ekt.oai.UnimarcCrosswalk">
   <constructor-arg>
     <value>
       http://www.loc.gov/MARC21/slim/
http://www.loc.gov/standards/marcxml/schema/MARC21slim.xsd
     </value>
   </constructor-arg>
</bean>
```

Example 3.
Specification of supported metadata formats

```
<bean id="virtual-sets" class="java.util.ArrayList">
  <constructor-arg>
    <list>
      <map>
        <entry key="name" value="dart" />
        <entry key="setSpec" value="dart" />
        <entry key="filters" value-ref="dart-filters"/>
      </map>
    </list>
  </constructor-arg>
</bean>

<bean id="dart-filters" class="java.util.ArrayList">
  <constructor-arg>
    <list>
      <ref bean="institution-filter"/>
      <ref bean="fulltext-filter"/>
      <ref bean="country-filter"/>
    </list>
  </constructor-arg>
</bean>
```

Example 4.
Declaration of virtual sets

by the BTE as well as the nature of BTE workflow which can embed modifiers and filters by providing just few lines of configuration XML code.

- To proof-check and validate the approach and the easiness of OAI harvesting via the implemented interface, we describe three systems with legacy data sources for which the tool was used.

5.2 Harvesting from a Z39.50 data source

The first implementation concerned material that the Technical Chamber of Greece (TEE) intended to contribute to Europeana, in particular collections that contain all their current publishing work (TEE digital library), some historical editions (1932-1980) and their multimedia content on engineers, buildings and posters. The descriptions of these objects are stored in the TEE bibliographic catalogue in the UNIMARC format, mixed with descriptions without online objects, which are inappropriate for Europeana. Additionally, their own content management system provides the above five collections together with other content, from their own regional subdivisions, their journal subscriptions, etc. The right selection of records has to be performed before they become available to Europeana. At the system level, the records could be made available through a Z39.50 programmatic interface.

In this use case, based on the enhanced OAI-PMH middleware mentioned before, we have developed a MARC/Z39.40 data loader capable of reading records from the Z39.50 server of TEE and a crosswalk to transform these MARC records to the ESE metadata schema of Europeana. Data loading was achieved using the JZKit open-source library[5] and the transformation of loaded records to meaningful records in BTE was based on the MARC4J tool[6]. The output of the transformation had the form of MARCXMLRecord objects (MARCXMLRecord is an abstraction for MARC records following the BTE Record interface). The second component that we had to implement was the crosswalk to provide the Records of BTE in the ESE metadata format of Europeana. A new crosswalk was written and its class was declared in the configuration file. The mapping of the internal fields that MARCXMLRecord uses to the ESE schema is done via the configuration file.

The aforementioned configuration did the bulk of the work; however, an important practical aspect concerned the controlled retrieval from the Z39.50 server, which was performed in chucks of maximum 100 records due to limited resources on the side of the TEE server. This issue could be resolved by BTE, as data loading can keep up until the requested number of records is returned from the legacy data source. The retrieval of the desired sets of records and the de-duplication were not possible using only queries (e.g. PQF or CQL) to the Z39.50 server, as the criteria for filtering were quite custom and complex (e.g. availability of full text that was specified in a non-standard way in the metadata records, filtering of records that are present in the database but are not published by the Technical Chamber of Greece, etc.). We implemented the set support using virtual filters and custom filters.

Furthermore, modifiers were injected in the BTE transformation workflow to transform records to the ESE format and perform various modifications to field values (e.g. normalisation; adjusting value encoding to Europeana standards). It is worth noting that development of filters and modifiers did not require any knowledge of the MARC and Z39.50 standards and the structure of MARC records.

The metadata records from the TEE collections that could be finally contributed to Europeana are approximately 6,800. The most frequent metadata field was dc:subject, which was usually repeated at least four times, and the 28,284 subjects that appear, contain 4,669 unique values. The lengthiest field is dc:title with 18 words on average and follows dc:description and dcterms:isPartOf with 15, while the dcterms:isPartOf is used in the 97 per cent of the records, and most fields are included once on each record.

5.3 Harvesting from a legacy database

Another case was that of the material of the Parthenon Frieze that has been digitised and scientifically documented by the Information and Education Department of the Acropolis Restoration Service (Hellenic Ministry of Culture). Initially, this material was published via an interactive web application implemented with Flash technology[7]. This application stored information in a custom relational database and did not provide an OAI-PMH-compliant programmatic interface. The OAI middleware that is mentioned in this paper was used for the development of a wrapper to expose the database contents as standard metadata records over OAI-PMH. A configurable data loader (SQLDataLoader) was implemented for retrieving information from relational databases and was used, with the appropriate parameters, for fetching data from the legacy database server. The BTE records created from this process did not resemble any standard metadata format; they had labels similar to the field names in the database. A couple of modifiers were needed to normalise certain data values in several fields and insert into each record additional Europeana-specific data fields. Finally, an output crosswalk was developed that produced records in the ESE schema. The mapping from the database schema to ESE was possible to be defined as an XML specification outside the source code of the output generator.

The source database and the transformation elements needed for this use case are detailed in the following.

The relational database describes the 119 stone blocks of the Parthenon Frieze. The main tables holding stone block metadata and their structure are shown in Table I (simplified for economy of presentation):

Each row in the first table holds information about the blocks in the Parthenon Frieze, whereas the second one holds the subjects of the representations depicted in the block.

The SQLDataLoader is a data loader that fetches one whole table and creates one record per row, based on a given mapping. For instance, the configuration at Example 5 instructs the data loader to fill the field "id" of the record with the data found in the column "bid" of the database table. The "fieldMap" property holds the mapping, which is expressed in XML and does not need software development skills for its definition.

After the record is created it is passed to the workflow for further processing. Four types of modifiers (SubjectModifier, ValueAddModifier, TitleModifier and LocationModifier) are applied to each record, as described below.

The subject modifier needs to get data from the database as well. In fact it reads the values from the table "Subjects" and adds a "subject" field in each stone block record (Example 6).

Table: BlockInfo *(each row represents a block)*	Attributes
	Bid
	Title
	Text
	Image
	Startposition
	Endposition
	Side
	Museum
Table: Subjects *(each row represents a subject)*	Attributes
	Sid
	Bid
	Subject

Table I.
Database tables holding the input metadata

```
<bean id="friezeLoader"
          class="gr.ekt.bteio.loaders.SQLDataLoader">
  <property name="db_connection"
    value="jdbc:mysql://example.server.com/parthenonfrieze_db"/>
  <property name="credentials"
    value="src/main/resources/credentials.txt"/>
  <property name="tableName" value="BlockInfo"/>
  <property name="fieldMap">
    <map>
      <entry key="bid"            value="id" />
      <entry key="title"          value="ttl" />
      <entry key="text"           value="txt" />
      <entry key="image"          value="contents" />
      <entry key="startposition"  value="start_pos" />
      <entry key="endposition"    value="end_pos" />
      <entry key="side"           value="side" />
      <entry key="museum"         value="museum" />
    </map>
  </property>
</bean>
```

Example 5.
Mapping specification for a relational database table

The “ValueAddModifier” (Example 7) is maybe the simplest modifier that can be written. It inserts a constant value to a given field. For example the “typeModifier” inserts the value “Sculpture” to the field “type”. This single modifier, with the appropriate configuration, is used to set the values of five different fields (type, format, medium, source, dataProvider and europeana.type) in the stone block metadata. The reason for the simplicity of this case is the fact that all stone blocks in the Parthenon Frieze have exactly the same value for all these fields.

The “titleModifier” and the “locationModifier” (Example 8) concatenate data from the record with constant string values to produce formatted values.

All the aforementioned modifiers were incorporated in the BTE workflow process by just adding them in the Spring XML configuration file as shown in Example 9.

Finally, as only the ESE output format should be supported, the corresponding BTE configuration (Example 10) is set to support only the ESE output crosswalk.

The ESE output crosswalk uses the data in the record to produce files suitable for ingestion to Europeana. A mapping between the internal BTE record and the OAI-PMH output is given in the listing in Example 11 (which is part of the Spring XML configuration).


```
<bean name="subjectModifier"
          class="gr.ekt.frieze.modifiers.SubjectModifier">
  <property name="db_connection"
    value="jdbc:mysql://example.server.com/parthenonfrieze_db"/>
  <property name="credentials"
    value="src/main/resources/credentials.txt"/>
  <property name="tableName" value="Subjects"/>
  <property name="fieldMap">
   <map>
    <entry key="subject" value="subject"/>
   </map>
  </property>
</bean>
```

Example 6.
Example of a subject modifier

```
<bean name="valueAddModifier"
        class="gr.ekt.frieze.modifiers.ValueAddModifier">
 <property name="fieldMap">

 <map>
     <entry key="type" value="Sculpture"/>
     <entry key="" value="image/jpg"/>
     <entry key="medium" value="Pentelic marble"/>
     <entry key="source" value="Acropolis Restoration Service"/>
     <entry key="europeana.dataProvider" value="National
            Documentation Centre (EKT)"/>
     <entry key="europeana.type" value="IMAGE"/>
   </map>
  </property>
</bean>
```

Example 7.
Specification of ValueAddModifier

```
<bean name="titleModifier"

          class="gr.ekt.frieze.modifiers.TitleModifier"/>

<bean name="locationModifier"

          class="gr.ekt.frieze.modifiers.LocationModifier"/>
```

Example 8.
Specification of titleModifier and locationModifier

In a second phase of development, a public DSpace repository was developed for the Parthenon Frieze material[8] and therefore an OAI-PMH server was available at the source system to expose the metadata to Europeana. The BTE was used in that case both for the initial loading and transformation of the legacy database contents to DSpace metadata records in Qualified Dublin Core, while the proposed enhanced OAI-PMH toolset was used to implement the mapping of the DSpace metadata to the ESE schema.

5.4 Harvesting from a raw XML file

The final use case of the BTE-enabled OAI-PMH server was that of the Hellenic Statistical Authority (EL.STAT). The ELSTAT digital library[9] is hosted by a custom-made software package that does not offer OAI-PMH support. This software is capable of exporting an XML document including all the records described in the MODS metadata format. Given this XML file (which is periodically updated at a specific network location), we were instructed to provide the records via the OAI-PMH protocol.

Using the BTE-enabled OAI-PMH server this was a trivial procedure and it is described in the following paragraphs.

At the very beginning, a pre-processing step of an XSLT transformation took place to transform the MODS XML file to Dublin Core format. This was judged to be necessary, as a DC output crosswalk was already implemented for other projects; however, the initial MODS XML file could be used as an input for the BTE.

```
<bean id="defaultTransformationWorkflow"
                                    class="gr.ekt.bte.core.LinearWorkflow" scope="prototype">
    <property name="process">
        <list>
          <ref bean="subjectModifier"/>
          <ref bean="valueAddModifier"/>
          <ref bean="titleModifier"/>
          <ref bean="locationModifier"/>
        </list>
    </property>
</bean>
```

Example 9.
Modifiers in a workflow specification

```
<bean id="crosswalks" class="java.util.HashMap">
  <constructor-arg>
    <map>
      <entry key="ese" value-ref="ese-crosswalk"/>
    </map>
  </constructor-arg>
</bean>

<bean id="ese-crosswalk"
  class="gr.ekt.oai.ESECrosswalk">
  <constructor-arg>
    <value>
      http://www.europeana.eu/schemas/ese/
                                    http://www.europeana.eu/schemas/ese/ESE-V3.4.xsd
    </value>
  </constructor-arg>
</bean>
```

Example 10.
Specification of supported crosswalks

Given the DC XML file, an XML data loader was developed to load the records in the system. The corresponding Spring XML is listed as Example 12.

The specified DataLoader created a specific type of BTE Records that each one holds an entire XML document as its primitive data. However, the OAI-PMH protocol specifies that each record must be associated with a timestamp to declare the creation or update date of the item. To overcome this issue, we added a modifier that adds a datestamp to each record (based on the date that the items were initially stored in our system), as shown in (Example 13).

Finally, regarding the output crosswalk to OAI_DC, we developed a new crosswalk which uses the XML document stored in each record to produce the OAI-PMH output (Example 14).

As far as the sets that will be exposed via the OAI-PMH protocol, as there are no native sets specified by "EL.STAT.", we can instruct BTE-enabled OAI server to provide one virtual sets of books (Example 15).

```
<bean id="dspace_output_spec"
    class="gr.ekt.bteio.specs.ESEOutputSpec">
  <property name="prefixDir" value="output"/>
  <property name="padding" value="5"/>
</bean>

<bean name="eseGenerator"
        class="gr.ekt.bteio.generators.ESEOutputGenerator">
  <constructor-arg>
    <map>
      <entry value="title"        key="dc.title" />
      <entry value="text"         key="dc.description" />
      <entry value="ttl"          key="dc.identifier" />
      <entry value="sideExt"      key="dcterms.isPartOf" />
      <entry value="medium"       key="dcterms:medium " />
      <entry value="format"       key="dcterms:hasFormat" />
      <entry value="subject"      key="dc.subject" />
      <entry value="type"         key="ese.type" />
      <entry value="source"       key="ese.provider" />
    </map>
  </constructor-arg>
</bean>
```

Example 11.
Example of mapping between internal representation and the output format (ESE)

```
<bean id="dataloader"
        class="gr.ekt.enhancedoaiserver.bte.ElstatXMLDataLoader"
        scope="prototype">
  <constructor-arg value="books_oai.xml"></constructor-arg>
</bean>
```

Example 12.
XML data loader configuration

```
<bean id="defaultTransformationWorkflow"
    class="gr.ekt.bte.core.LinearWorkflow" scope="prototype">
  <property name="process">
    <list>
      <ref bean="datestamp-modifier"/>
    </list>
  </property>
</bean>

<bean id="datestamp-modifier"
    class="gr.ekt.enhancedoaiserver.bte.ElstatDatestampModifier">
  <property name="datestamp" value="2014-01-31T09:56:58Z">
  </property>
</bean>
```

Example 13.
Datestamp modifier configuration

As can be seen, no filters are defined for the specific sets, as we wanted all the records to be available under the "books" set.

As a result of the aforementioned work, the ELSTAT digital library material has been successfully incorporated in the openarchives.gr aggregator at the metadata level (http://openarchives.gr/organisations/view/54).

6. Summary and future work

Small organisations need tools that can serve them to perform metadata transformation tasks, without re-implementing functionality available elsewhere, and to participate to aggregator efforts in a more straightforward and flexible manner according to their own collection set-up and requirements.

They need tools that only accept an easy configuration, without requiring programming skills, to convert metadata to specific syntax and schema, to select records according to predefined rules, to enrich the metadata and to provide the desired metadata elements. We designed and implemented such tools for efficient OAI-PMH exchange of metadata. With the proposed approach, our OAI-PMH server can apply advanced logic for selective harvesting such as transformations among different formats and schemata, filtering and updating of data. Content providers can define dynamic sets to convert their metadata and the corresponding schema mappings, without altering their collections and schemas. Even when their software does not support OAI-PMH, they can use our modular implementation that enables retrieval of metadata records from a variety of non-OAI-PMH sources.

```
<bean id="crosswalks" class="java.util.HashMap">
  <constructor-arg>
    <map>
      <entry key="oai_dc" value-ref=" oaidc-crosswalk"/>
    </map>
  </constructor-arg>
</bean>

<bean id=" oaidc-crosswalk"
    class="gr.ekt.enhancedoaiserver.bte.ElstatOAIDCCrosswalk">
  <constructor-arg>
    <value>http://www.openarchives.org/OAI/2.0/oai_dc/
             http://www.openarchives.org/OAI/2.0/oai_dc.xsd</value>
  </constructor-arg>
</bean>
```

Example 14.
Configuration of the utilised crosswalk

```
<bean id="virtual-sets" class="java.util.ArrayList">
  <constructor-arg>
    <list>
          <map>
            <entry key="name" value="Book" />
            <entry key="setSpec" value="Book" />
            <entry key="filters" value-ref="book-set-filters"/>
          </map>
    </list>
  </constructor-arg>
</bean>

<bean id="book-set-filters" class="java.util.ArrayList">
  <constructor-arg>
    <list>
    </list>
  </constructor-arg>
</bean>
```

Example 15.
Configuration of virtual sets

We presented here three cases in which the tools were applied: retrieving data from a library catalogue through the Z39.50 protocol, retrieving data from a legacy database and exposing OAI-PMH records out of a raw XML export. All these systems were not designed to provide data using the OAI-PMH protocol, and needed to be redesigned to provide the desired records to any metadata aggregator.

Further work is being planned along various paths. The case studies provided clear indications that the proposed approach leads to very good performance both in terms of harvesting speed and consumption of computing and memory resources. A detailed investigation of performance issues is an interesting extension of the present work. Other plans include the design of an OAI-PMH proxy that can apply the configured operations over a legacy OAP-PMH server, the incorporation of the developed modular tools into various open-source OAI-PMH servers and the application of the proposed approach with more content providers and a systematic user study to capture their experiences with the tools in terms of utility and ease of configuration and extension.

An earlier version of this paper was presented at the 3rd International Conference on Integrated Information, IC-ININFO, held in Prague, Czech Republic, from 5 to 9 September, 2013, http://history.icininfo.net/2013/.

Notes

1. Open Archives Initiative – Protocol for Metadata Harvesting.
2. VOA3R EU project, available at: www.voa3r.eu (accessed 8 March 2014).
3. BTE: https://github.com/EKT/Biblio-Transformation-Engine (accessed 8 March 2014).
4. DSpace: Digital Repository Platform, available at: www.dspace.org/ (accessed 8 March 2014).
5. JZKit open-source library: www.k-int.com/jzkit
6. MARC4J tool: http://marc4j.tigris.org/
7. Parthenon Frieze interactive: www.parthenonfrieze.gr
8. DSpace repository for Parthenon Frieze material: http://repository.parthenonfrieze.gr
9. ELSTAT digital library: http://dlib.statistics.gr/

References.

Banos, V. (2009), "Open archives engine software", available at: http://vbanos.gr/blog/2010/06/19/open-archives-engine/ (accessed 20 June 2014).

Banos, V. (2010), "DSpace plugin for Europeana Semantic Elements (ESE)", available at: http://el.vbanos.gr/blog/2010/02/02/dspace-plugin-for-europeana-semantic-elements-ese/ (accessed 20 June 2014).

Banos, V. (2011), "Open archives initiative protocol for metadata harvesting validation and data extraction tool", available at: http://oaipmh.com (accessed 11 February 2014).

EuropeanaLocal (2008), "EuropeanaLocal", available at: www.europeanalocal.eu/ (accessed 11 February 2014).

Freire, N. and Reis, D. (2009), "Guidelines for preparing a Z39.50/SRU target to enable metadata harvesting", Deliverable D-2.3, Project TELplus: The European Library Plus, Project reference: ECP-2006-DILI-510003, available at: http://cyberdoc.univ-lemans.fr/PUB/CfU/Journee_UNIMARC_Lyon/TELplus-D2.3_v1.0%5B1%5D.pdf (accessed 21 June 2014).

Garoufallou, E. and Asderi, S. (2010), "Digital libraries and the digital working environment: what is their effect on library staff for sharing their knowledge?", in Katsirikou, A. and

Skiadas, C. (Eds), *New Trends in Qualitative and Quantitative Methods in Libraries, 2nd Qualitative and Quantitative Methods in Libraries, Proceedings of the International Conference Chania*, World Scientific, Greece, pp. 359-365.

Garoufallou, E., Asderi, S. and Koutsomiha, D. (2010), "Digital libraries as knowledge management systems", *International Scientific Conference, eRA 5: The SynEnergy Forum: The Conference for International Synergy in Energy, Environment, Tourism and contribution of Information Technology in Science, Economy, Society and Education, Piraeus, Greece*, *15-18 September*.

Garoufallou, E., Banos, V. and Koulouris, A. (2013), "Solving aggregation problems of Greek cultural and educational repositories in the framework of Europeana", *International Journal of Metadata, Semantics and Ontologies (IJMSO)*, Vol. 8 No. 2, pp. 134-144.

Giannakopoulos, G., Manesi, D.K. and Zervos, S. (2012), "Approaching information as an integrated field: educating information professionals", in Giannakopoulos, G.A. and Sakas, D.P. (Eds), Integrated Information. *International Conference on Integrated Information*, I-DAS, *Piraeus, Kos, Greece*, *29 September-3 October*, pp. 128-131.

Houssos, N., Stamatis, K., Banos, V., Kapidakis, S., Garoufallou, E. and Koulouris, A. (2011), "Implementing enhanced OAI-PMH requirements for Europeana", in Gradmann, S., Borri, F., Meghini, C. and Schuldt, H. (Eds), *Proceedings of the International Conference on Theory and Practice of Digital Libraries (TPDL 2011), Lectures Notes in Computer Science (LNCS)*, Vol. 6966, Springer-Verlag, Berlin Heidelberg, Berlin, *25-29 September*, pp. 396-407.

IRN Research (2011), "Europeana – online visitor survey", Research report version 3, available at: http://pro.europeana.eu/c/document_library/get_file?uuid=334beac7-7fc2-4a4e-ba23-4dcc1450382d&groupId=10602 (accessed 11 February 2014).

Koninklijke Bibliotheek (2009), "Europeana", available at: www.europeana.eu (accessed 11 February 2014).

Konstantinou, N., Houssos, N. and Manta, A. (2014), "Exposing bibliographic information as linked open data using standards-based mappings: methodology and results", *Procedia Social and Behavioral Sciences (in press)*, Prague, Czech Republic.

Mazurek, C., Mielnicki, M. and Werla, M. (2005), "Selective harvesting of regional digital libraries and national metadata aggregators", *9th ACM/IEEE-CS Joint Conference on Digital libraries (JCDL 2009)*, New York, NY, pp. 429-430.

Mazurek, C., Mielnicki, M., Parkola, T. and Werla, M. (2009), "The role of selective metadata harvesting in the virtual integration of distributed digital resources", *ENRICH Final Conference*, *Spain, Madrid*, pp. 27-31.

Ntonas, K. and Kokkoras, F. (2007), "DEiXTo", available at: www.deixto.com (accessed 8 March 2014).

Rowlatt, M., Davies, R. and Komen, L. (2011), "EuropeanaLocal: it's objectives, activities and impact", Project presentation: results D1.11, available at: www.europeanalocal.eu/eng/Document-Library/Project-Deliverables (accessed 11 February 2014).

Sanderson, R., Young, J. and LeVan, R. (2005), "SRW/U with OAI: expected and unexpected synergies", *D-Lib Magazine*, Vol. 11 No. 2, available at: www.dlib.org/dlib/february05/sanderson/02sanderson.html (accessed 8 March 2014).

Sidhunata, H.R., Croucher, J. and Frances, M. (2011), "Selective harvesting: creating and ingesting custom OAI-PMH sets", *4th eResearch Australasia Conference*, *Gold Coast*, *November*.

Stamatis, K., Konstantinou, N., Manta, A., Paschou, C. and Houssos, N. (2012), "Biblio-transformation-engine: an open source framework and use cases in the digital libraries domain", *7th International Conference on Open Repositories, Edinburgh, George Square Campus, Edinburgh*.

The Europeana Office (2012), "Europeana semantic elements specification: version 3.4.1", available at: http://pro.europeana.eu/documents/900548/dc80802e-6efb-4127-a98e-c27c95396d57 (accessed 11 February 2014).

University of Minho (2011), "OAI Extended AddOn", available at: http://projecto.rcaap.pt/index.php/lang-en/consultar-recursos-de-apoio/remository?func=fileinfo&id=337 (accessed 8 March 2014).

Vassilakaki, E. and Garoufallou, E. (2013), "Multilingual digital libraries: a review of issues in system-centered and user-centered studies, information retrieval and user behaviour", *International Information and Library Review*, Vol. 45 Nos 1/2, pp. 3-19.

Further reading

Devarakonda, R., Palanisamy, G., Green, J.M. and Wilson, B.E. (2011), "Data sharing and retrieval using OAI-PMH", *Earth Science Informatics*, Vol. 4 No. 1, pp. 1-5.

Koulouris, A., Garoufallou, E. and Banos, E. (2010), "Automated metadata harvesting among Greek repositories in the framework of EuropeanaLocal: dealing with interoperability", in Katsitikou, A. and Skiadas, C. (Eds), *New Trends in Qualitative and Quantitative Methods in Libraries, 2nd Qualitative and Quantitative Methods in Libraries, Proceedings of the International Conference on QQML 2010, World Scientific, Chania*, pp. 331-337.

About the authors

Nikos Houssos is an Associate Research Scientist and Head of Software Applications Development Unit at the National Documentation Centre/National Hellenic Research Foundation. He is the software architect of the Greek "National Information System on Research and Technology", a national-scale scholarly communications e-infrastructure. He has participated in various European Union and national projects in the areas of digital libraries, data management and mobile communications and services. He is an elected Board member of euroCRIS since 2009. He holds a PhD in computer science (2004) from the University of Athens and has served (2004-2007) as an Adjunct Lecturer at the Technical University of Crete. He has co-authored more than 30 publications in international scientific journals and conferences. Nikos Houssos is the corresponding author and can be contacted at: nhoussos@ekt.gr

Kostas Stamatis received his MEng in electrical and computer engineering from National Technical University of Athens in 2002. He also holds an MSc in information networking from Carnegie Mellon University since 2004. He is working for National Documentation Center in Greece for several years on developing digital repositories and libraries and tools for data cleansing and clustering. In the past, he has been employed by the IT company of informatics and research, Athens Information Technology, as a researcher participating in various European programmes regarding the interaction between humans and machines as well as the image and video processing concepts. He is a member of the open-source digital repository software DSpace committer's group.

Panagiotis Koutsourakis is a software developer at the National Documentation Centre of Greece. In the past he has worked as Assistant Researcher at École Centrale Paris, France, and the University of Crete, Greece. He holds a BSc and an MSc in computer science from University of Crete, Greece.

Sarantos Kapidakis is Professor at the Department of Archive, Library and Museum Sciences, and Dean of the Faculty of Information Science and Informatics at Ionian University, Corfu, Greece, and Director of the Laboratory on Digital Libraries and Electronic Publishing. He is also a member of the Steering Committee of the National Archives of Greece. In the past, he has been employed by the National Documentation Centre, Greece; MIT, USA; the University of Crete,

Greece; and the Foundation for Research and Technology – Hellas. He received a PhD degree in computer science from Princeton University in 1990. He also holds an MSc from Princeton University and a Diploma in electrical engineering from the National Technical University of Athens. As part of his research on digital libraries, he participated in the DELOS Network of Excellence on Digital Libraries, and was the Chair of the European Conference on Digital Libraries in 2009.

Emmanouel Garoufallou is a Lecturer at the Department of Library Science and Information Systems and Project Manager of the programme "Open Source Digital Library Services of Alexander TEI of Thessaloniki" at the Alexander Technological Educational Institute of Thessaloniki (ATEITh), Greece. He is also coordinator of ATEITh libraries. He is Project Manager and Research Associate of various projects of the Veria Central Public Library (the Award-winning Access to Learning Award of the Bill and Melinda Gates Foundation) such as the AccessIT Plus, Light, Entitle, EuropeanaLocal. He served as Chair of the 7th MTSR2013, as Programme Chair of the 8thMTSR2014 Conference and as member of the MTSR steering committee. He currently serves as an editorial board member of various international journals such as the *Education for Information* and *International Information and Library Review*, and as associate editor of the *Program: Electronic Library and Information Systems* journal.

Alexandros Koulouris is Lecturer in the Department of Library Science and Information Systems at the Technological Educational Institute of Athens. He is member of the Europeana Network (formerly CCPA) and of the Laboratory on Digital Libraries and Electronic Publishing at Ionian University and collaborator of the Veria Central Public Library. He has participated in various EC programs, such as DELOS, EuropeanaLocal, etc. In the past, he has worked as a librarian for the National Technical University of Athens and for the National Documentation Centre of Greece. He holds a PhD in information science from Ionian University, a BA in library science from the TEI of Athens and a BA (Hon) in international and European studies from Panteion University. More can be found on his website http://users.teiath.gr/akoul.