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

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

National Documentation Centre / National Hellenic Research Foundation, Greece

7th International Conference on Open Repositories (OR2012), Edinburgh, 9-13 July 2012

Agenda

- Introduction
- Motivation - the recurring need for data transformations
- The proposed solution
- Use cases / experience reports
- Summary – conclusions – future work

Motivation

- Data transformations are needed everywhere in digital libraries / scholarly communication systems
- Painful and tedious procedure
- Many sub-tasks of the entire procedure reoccur and could be reused
- Need for systematic framework for data transformations to accelerate the process, reduce errors and facilitate reuse

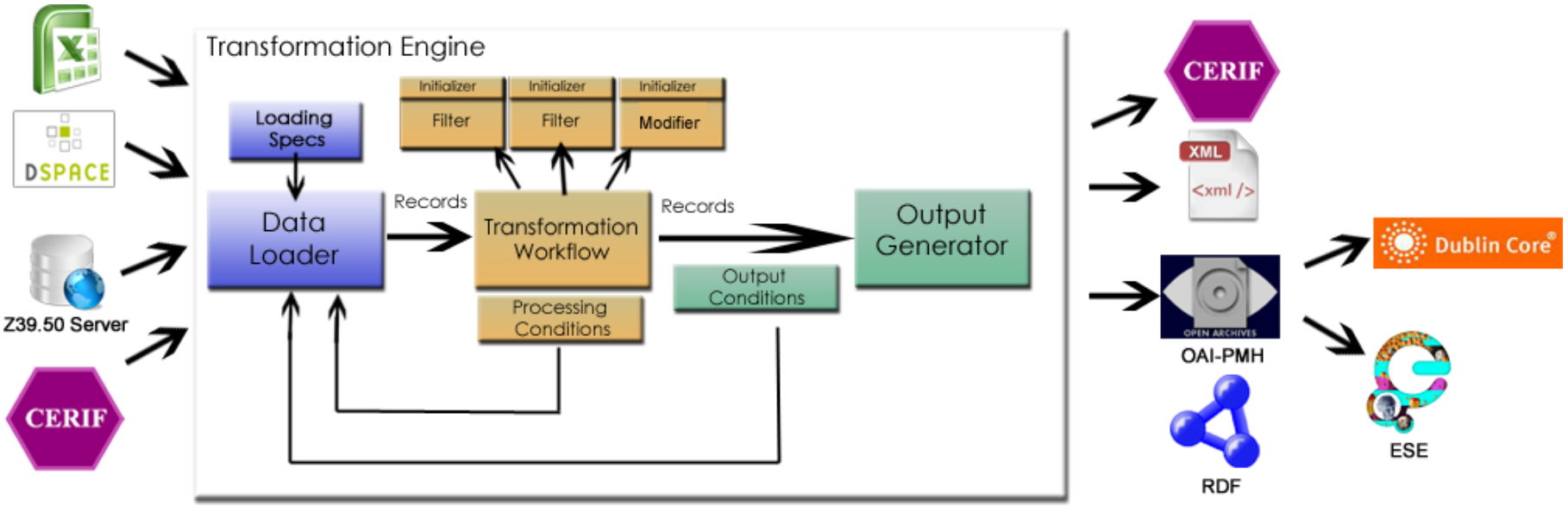
Analysis – basis steps in data transformations

- Retrieve source data records
- Apply processing:
  - (Optionally) Remove data records
  - (Optionally) Add/modify/delete field values within records
  - Transform data source to output format (implement the corresponding mapping)
- Generate desired output
  - Export to a file and/or directly update databases / external systems
- Need for incremental / selective data loading -> processing and output conditions may require repeated execution of the loading/processing cycle

Design goals

- Customisable, non-intrusive, easy to use, integrate and extend (e.g. support a variety of data source types)
- Separation of concerns in development – e.g. development of transformation logic independent of data sources
  - Example: No need to be aware of MARC to develop a function to harmonise encoding of dates
- Support for recurring execution of the data loading/processing cycle according to specific criteria (e.g. useful for OAI-PMH)

The biblio transformation engine



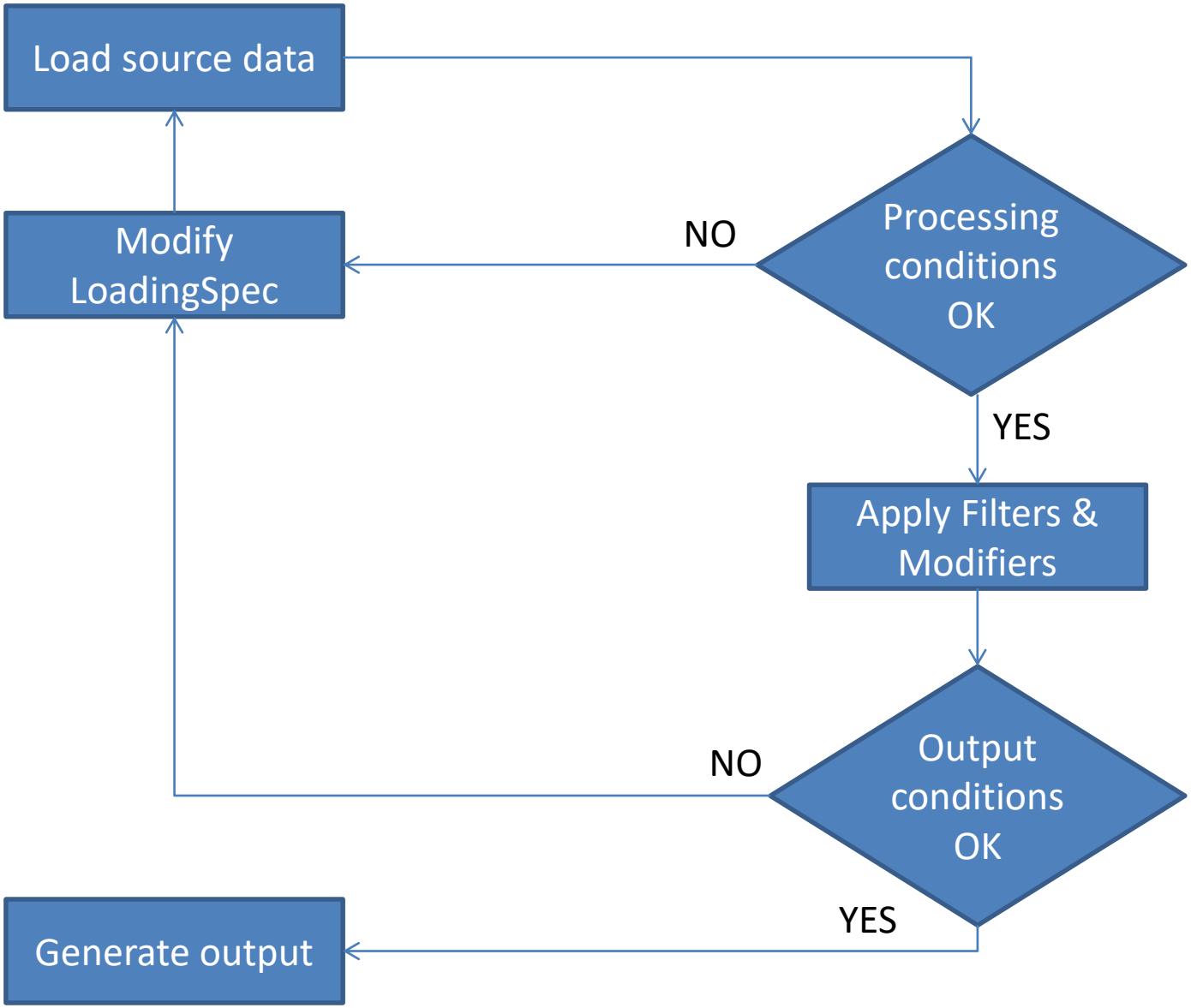
Components of the engine

- ***Data Loader***: Retrieves data from input source(s) according to ***DataLoading Spec***
- ***ProcessingStep***: Transforms input in some way
  - ***Filters***: removes records according to specific criteria
  - ***Modifier***: updates records according to specific criteria
  - ***Initializer:*** initializes data in processing steps (e.g. load author names to Filter)
- ***Output Generator***: Creates the desired output (e.g. export file, direct update of database)
- ***Record*** abstraction: simple common interface for all types of records that allows complex transformation functions

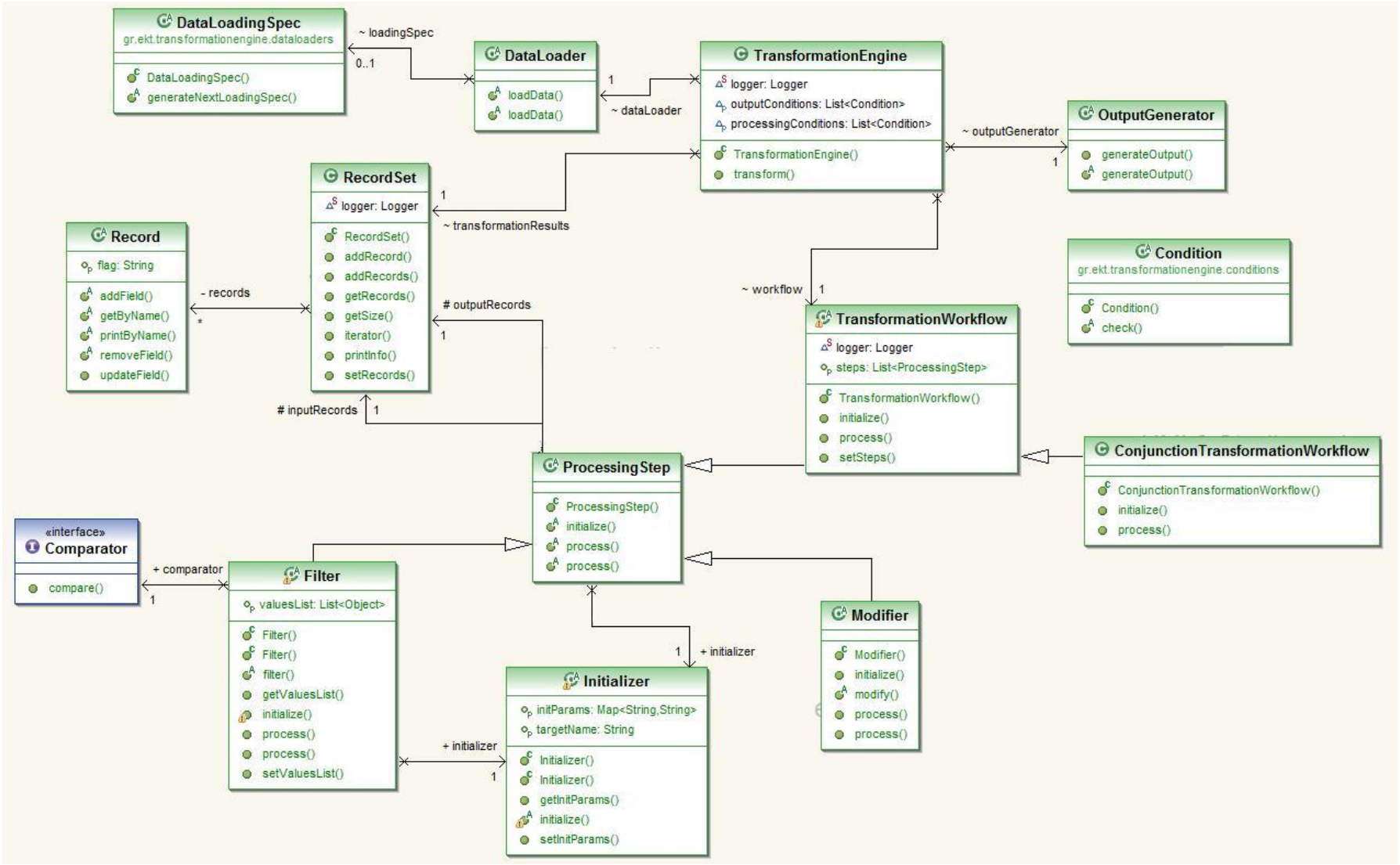
Processing workflow

- Load data – transform input to records
- If processing conditions are met, begin processing – sequential execution of Filters and Modifiers
- If output conditions are met, begin output – execution of OutputGenerator(s)

Processing workflow



The transformation engine – data model



Implementation

- FLOSS library developed in Java (maven used as a build tool)
- Configuration outside the code - dependency injection mechanisms of the Spring framework core container
  - Specification of Data Loader, Processing Steps, Conditions, OutputGenerator
  - Mapping from source to target format (for one-to-one field mappings)

Example of mapping configuration

```
<?xml version="1.0" encoding="UTF-8"?>
<keymapping>
    <map>
        <!-- Title -->
        <key_before>TI</key_before>
        <key_after>dc.title</key_after>
    </map>
    <map>
        <!-- Author -->
        <key_before>AU</key_before>
        <key_after>dc.contributor.creator</key_after>
    </map>
    <map>
        <!-- Publication Title -->
        <key_before>PT</key_before>
        <key_after>dc.type</key_after>
    </map>
```

FLOSS library

- Available at
  http://code.google.com/p/biblio-transformation-engine/
- European Union Public License
- Feel free to download and use it!
- Looking forward to feedback, questions,… (contributions also welcome ☺)

Use case 1 – Generate Linked Open Data

- Sources: Repository records, legacy cultural material records, research information in CERIF
- Corresponding data loaders reused
- Filters/Modifiers can be totally agnostic of RDF and input formats
- Use Jena RDF library to generate RDF triples
- Add/generate appropriate identifiers/URI for each entity (either at the modifier or output generator level)

Use case 2 – Import/export data/export to/from repositories

- Source record formats: EndNote, RIS, Bibtex, UNIMARC
- Developed data loaders for each format, re-used output generator for DSpace
- Export to different formats and reference styles based on repository records
  - Implemented for DSpace
  - For reference styles uses the citeproc-js library and the Citation Style Language (CSL)

Use case 3 – Feed the VOA3R aggregator

- Get records of the Hellenic National Archive of Doctoral Dissertation (HEDI – didaktorika.gr) to the VOA3R aggregator (Virtual Open Access Agriculture & Aquaculture Repository)
- Developed subject-based filter and injected it into an enhanced OAI-PMH server using the library.
- ~1070 of approximately 23.500 records, needed to apply techniques to cater for the distribution sparsity of “suitable” records combined with resumption token
- Seamless on-the-fly deployment and co-existence with sets targeted to other aggregators (DART, openarchives.gr)

Use case 4 – Feed Europeana

- Include in Europeana content from the Technical Chamber of Greece (TEE)
- Records in TEE library catalog (UNIMARC), available through a Z39.50 interface
- Developed Z39.50 data loader, appropriate filters and modifiers (independent of UNIMARC)
- Mapping to ESE implemented through a modifier
- ~6800 from the TEE records sent to Europeana
- Repeatable, automated procedure through an enhanced OAI-PMH server using the library

Future work

- Support more types of data transformations (contributions welcome ☺)
- Extend declarative specification of mappings to cover more sophisticated cases
- Configurable support for common data model to facilitate reuse of Filter and Modifier implementations
- Systematically study the user experience, identify and implement potential improvements

Thank you for your attention!

- More info:

http://code.google.com/p/biblio-transformation-engine/

kstamatis AT ekt.gr

nkons AT ekt.gr

amanta AT ekt.gr

cpaschou AT ekt.gr

nhoussos AT ekt.gr