Project 1

Enterprise Application Integration (MIEBIOM)
Department of Informatics Engineering
Delivery date: 25 October 2023

Objectives

- Learn XML and how to process XML using JAXB
- Learn XML Schema and XSLT

Final Delivery

- You must submit your project in a zip file using Inforestudante. Do not forget to associate your work colleague during the submission process.
- The submission contents are:
  - All created code (and XML data) necessary to compile and run the requested applications.
- After submitting, you are required to register the (extra-class) effort spent solving the assignment. This step is mandatory. Please fill the effort form at:
  https://docs.google.com/spreadsheet/viewform?formkey=dHFha3NuWE1pYWJDbTJMTWcxTWRMS1E6MA

Grading

- Overall quality of the data model used for representing data (XML/XSD);
- Quality of the code (organization, modularity, formatting, code conventions, etc.);
- Simplicity of the solution;
- Final presentation of the work.

Resources

Software:

- **Trang tool** – jar file download at https://mvnrepository.com/artifact/org.relaxng/trang/20220510
- **JAXB Libraries** (compiler under the bin/ directory)
  - https://eclipse-ee4j.github.io/jaxb-ri/
  - https://repo1.maven.org/maven2/com/sun/xml/bind/jaxb-ri/4.0.3/jaxb-ri-4.0.3.zip

Documentation:

- **XML** Technologies – http://www.w3schools.com/xml
  - Check the Schema; XPath; XSLT links
- **JAXB Tutorial** – http://jaxb.java.net/tutorial/index.html
- **HTML** basic information – http://www.w3schools.com/html/

Project Description

In this project, you will be using XML technologies and techniques for reading, validating, processing, and writing XML documents. This will be accomplished using the Java programming language, JAXB, XML Schema, and XSLT.

In this assignment, you will create a set of applications that integrate, process, and categorize information available about television series extracted from an on-line software system. The project includes the following parts:

- **Part I – Selector**: Reads an XML document holding information about *researchers (e.g., name, affiliation, ...) and publications (e.g., paper title, conference name, …),* then validates it and processes the corresponding data, to write a shorter XML document (based on user preferences).
- **Part II – Processor:** Converts the XML document, produced in Part I, to another XML document (which essentially adds some information).
- **Part III – HTMLViewer:** Converts the XML document, produced in Part I, to HTML using XSLT.

Figure 1 shows the overall scenario of this project. The three parts that compose the project are described in the following paragraphs.

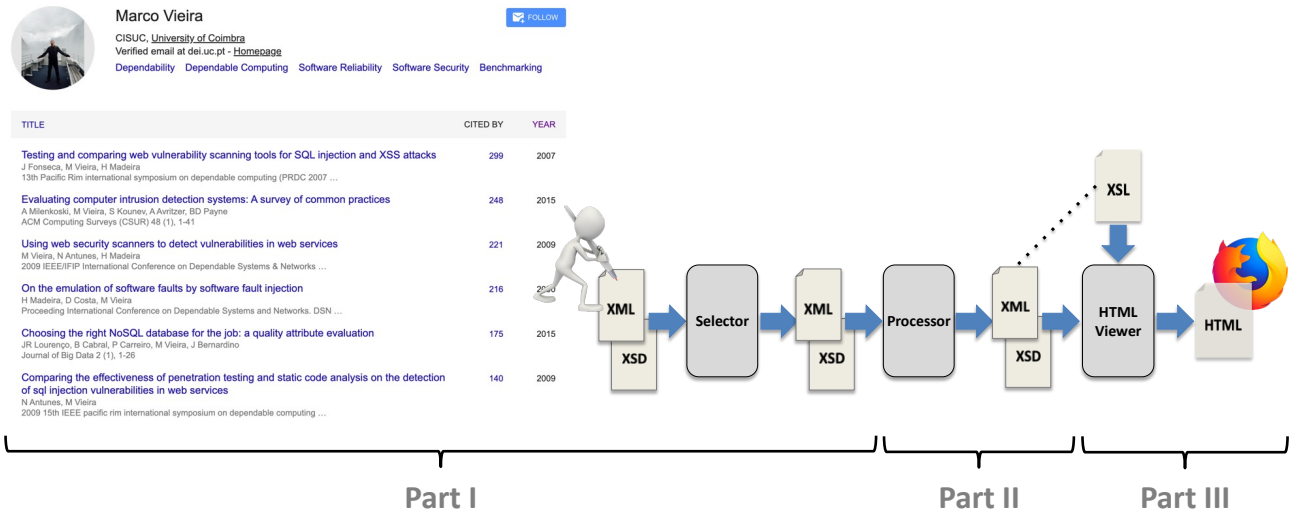

Figure 1 – The information flow

Part I – Selector

The source of information for this project is **Google Scholar**[1]. In Part I, you will create an application, named *Selector*, that reads information in XML format (this information is created by yourself, but is based on the information present on the website) and produces a shorter version (also in XML), based on user preferences.

You should start by analyzing what is the important information to retrieve and create a sample XML file (based on the information provided by the site) to be later processed by

[1] https://scholar.google.ca/citations?hl=en&user=l3cF0LoAAAAJ&
Examine a few pages, such as this one, to understand the overall structure and content, to be able to define the right data model.

the *Selector*. All XML files defined/produced (i.e., the application input and output) should have an associated XML Schema. The XML Schemas allow you to generate Java classes that can be used by an application to hold and manipulate the information.

In practice, to create the *Selector* application, you can do the following:

- Add the JAXB libraries to your project
- Define the XML Schema(s). This may involve the use of *trang* to create XSD from XML. Verify and tune your generated Schema(s).
- For each XML Schema, generate the respective Java classes using the JAXB compiler, named *xjc*.
- Once you have the Java classes that can keep the data, you can instantiate and use them normally in your code.

Each time the *Selector* runs, it reads and validates the input XML, filters the data (i.e., discards some items, according to specific user rules) and then produces XML output. *The rules must target (at least) the host institution and the researcher interests (generally visible at the top of the page). For example, a user may specify a set of rules to select researchers from the University of Coimbra and/or with interests on biomedical engineering, web services, or security.* All other cases will be excluded from the application output.

Part II – Processor

This application generates basic statistics using the information produced by the *Selector*. You may define your own statistics, but they must include at least the total number of researchers processed, total number of conferences/journals, the total number of citations considering all researchers, and the top N publications holding the highest citation count. The output of this application will include both the statistics and the *Selector* output data.

Part III – HTMLViewer

In this part of the project, you will define an XSL Transformation for converting the XML file produced by the *Processor* to HTML. You may use Java code directly in the processor application to generate the HMTL.