# IMY 210
# Semester Project 2011

## The Scenario

You were recently hired by a web development company as the resident XML expert person for their projects. Your superior decided to put you on a job for one of the company's more important clients: an online Blu-ray store specialising in animation movies, called, quite originally: Online Store for Blu-ray Animations (OSBA).

OSBA wants to send product information to their customers in a variety of ways, and they want you to handle it.

## Your Mission

It is a good idea to structure the information in such a way that it can be transformed easily into various formats, thus streamlining the maintenance of the online store and keeping subscribers up to date via newsletters. OSBA also wants to store pertinent product data in a relational database for easy access, so you must ensure that the data is transformed into a format that can be imported easily by such databases.

You must implement your solution by structuring and transforming the product information using XML and XSL.

## Specifications

### Provided Files

- A list of products currently available on OSBA, compiled and reworked into an XML document called "available.xml".
- The critical reception of the available products, compiled into a single XML document called "reception.xml".
- Two film distributors have given OSBA their XML documents that contain information on their upcoming releases. Each distributor used their in-house XML structures for their documents. These files are: "numetro.xml" and "sterkinekor.xml".

- A collection of images for use in the project. Each available release and upcoming product has cover art, and each available release has three still images. A logo for OSBA is also included.

- A collection of batch files (.bat) that you must use to test your project: one file that will open the command prompt in the current folder, one to combine all XML documents into a single document, and one for each transformation to perform.

- A command-line XSLT processor, MSXSL, which is necessary for creating actual result files instead of displaying the result directly in the browser.

**Tasks**

1. **Transform all XML documents into a single master XML document**

- You have: "numetro.xml", "sterkinekor.xml", "available.xml", and "reception.xml".

- To transform the information contained in these XML documents into several displayable formats, you need to combine these documents into a single well-structured XML document.

- Create an XSLT style sheet, "osba.xsl", to transform the source XML documents into a single new XML document, "osba.xml".

    - Structure the result document in the following way:

        - Create an appropriate root element.

            - Create a child for available releases.

                - Copy the XML code from "available.xml" here.

                - Add an element to each release's structure for that release's critical reception. Copy the reception XML code from "reception.xml." (Do not first list all releases and then list all receptions.)

            - Create a child for upcoming releases.

                - Since the distributors' in-house XML structures do not follow XML best practice guidelines, design a new and improved standard structure for all upcoming releases across all distributors. (A standard structure means that each upcoming release must be marked up the same way.)

    - Create a namespace for each distributor: one for Nu Metro and one for Ster Kinekor.

    - Add the upcoming releases to the XML file using the new structure, retrieving data from "numetro.xml" and "sterkinekor.xml". Place each release into their distributor's namespace.

- Things to look into that we didn't cover in class:

    - Look for an XSLT instruction that will allow you to copy XML code from another XML file without having to use <xsl:value-of> for everything.

    - Look for an XSLT function (not an XPath function) that will allow you to use multiple XML documents as a source inside a single style sheet. Your project needs to create the result XML document using a single transformation.

## 2. Create schema(s) for your master XML document

- You need to validate the structure of your master XML document, as well as populate the namespaces contained within.

- You must create your schema(s) using XML Schema, not DTD.

## 3. Create an HTML newsletter from the OSBA data

- You now have a valid and well-formed "osba.xml".

- OSBA wants an information newsletter in HTML format, which lays all their information out in a visually appealing way. They can then e-mail this to their subscribers.

- Create a style sheet, "style1.xsl", that must transform "osba.xml" to "output1.html". Wherever possible, data must be retrieved from the source XML, not written directly into the style sheet.

- OSBA has provided their logo image, which you should display at the top of the HTML page.

- Add a greeting message at the top and a farewell message at the bottom, in typical e-mail fashion.

- The HTML page must display all upcoming release, available release, and critical reception information. An available release's critical reception should appear together with its release information, instead of first displaying all available releases, and then displaying all receptions.

- You must make use of all images provided for each release, placed appropriately.

- Put some effort into laying out the information in an organised manner, as well as making the page visually appealing through fonts, colours, and so on. Part of the mark will go to overall visual and layout impression.

### 4. Create a plain-text newsletter from the OSBA data

- You need "osba.xml" for this task.

- OSBA also needs a simplified TXT version of their newsletter for e-mail clients that cannot display HTML.

- You need to create a style sheet, "style2.xsl", that must transform "osba.xml" to "output2.txt". Wherever possible, data must be retrieved from the source XML, not written directly into the style sheet.

- The result TXT document represents the body of an e-mail that will be sent to subscribers.

- Needless to say, no image information must be present.

- Add the same greeting and farewell message that you used in the HTML newsletter.

- Add all upcoming release and available release information. For each available release, provide **ONLY** the first sentence of the first paragraph of its critical reception. Show it with the release; do not group all releases' reception information together by itself.

- Lay the information out in an easy-to-read manner. Part of the mark will go to overall layout impression.

### 5. Create a PDF newsletter from the OSBA data

- You need "osba.xml" for this task.

- OSBA wants to cover all their bases and subsequently reach as many people as possible in the hope of taking over the world (at least as far as Blu-ray sales are concerned). Thus, they also wish to print out fancy PDF newsletters that they will stuff into unsuspecting mailboxes.

- You need to create a style sheet, "style3.xsl", that must transform "osba.xml" to "output3.fo", which will be formatted to "output3.pdf". Wherever possible, data must be retrieved from the source XML, not written directly into the style sheet.

- The PDF document must display all upcoming release, available release, and critical reception information. An available release's critical reception should appear together with its release information, instead of first displaying all available releases, and then displaying all receptions.

- Create a cover page that displays the OSBA logo, as well as the same greeting message used in the HTML and TXT newsletters. Also display the farewell message on the last page of the PDF newsletter.

- Make use of all images provided for each release, placed appropriately.

- Present the information in a well-organised manner. The visual aspects of the newsletter should be consistent with the HTML version. Part of the mark will go to overall visual and layout impression.

## 6. Create a database-ready CSV file from the OSBA data

- You need "osba.xml" for this task.

- To make searching functionality on their website easier, OSBA wants to store pertinent information in a relational database. They need this information in CSV format, so that they can import it into their database.

- A CSV file is a text file with a special format. See this Wikipedia page: http://en.wikipedia.org/wiki/Comma-separated_values, for more information and an example of what your CSV file should look like.

- You need to create a style sheet, "style4.xsl", that must transform "osba.xml" to "output4.csv".

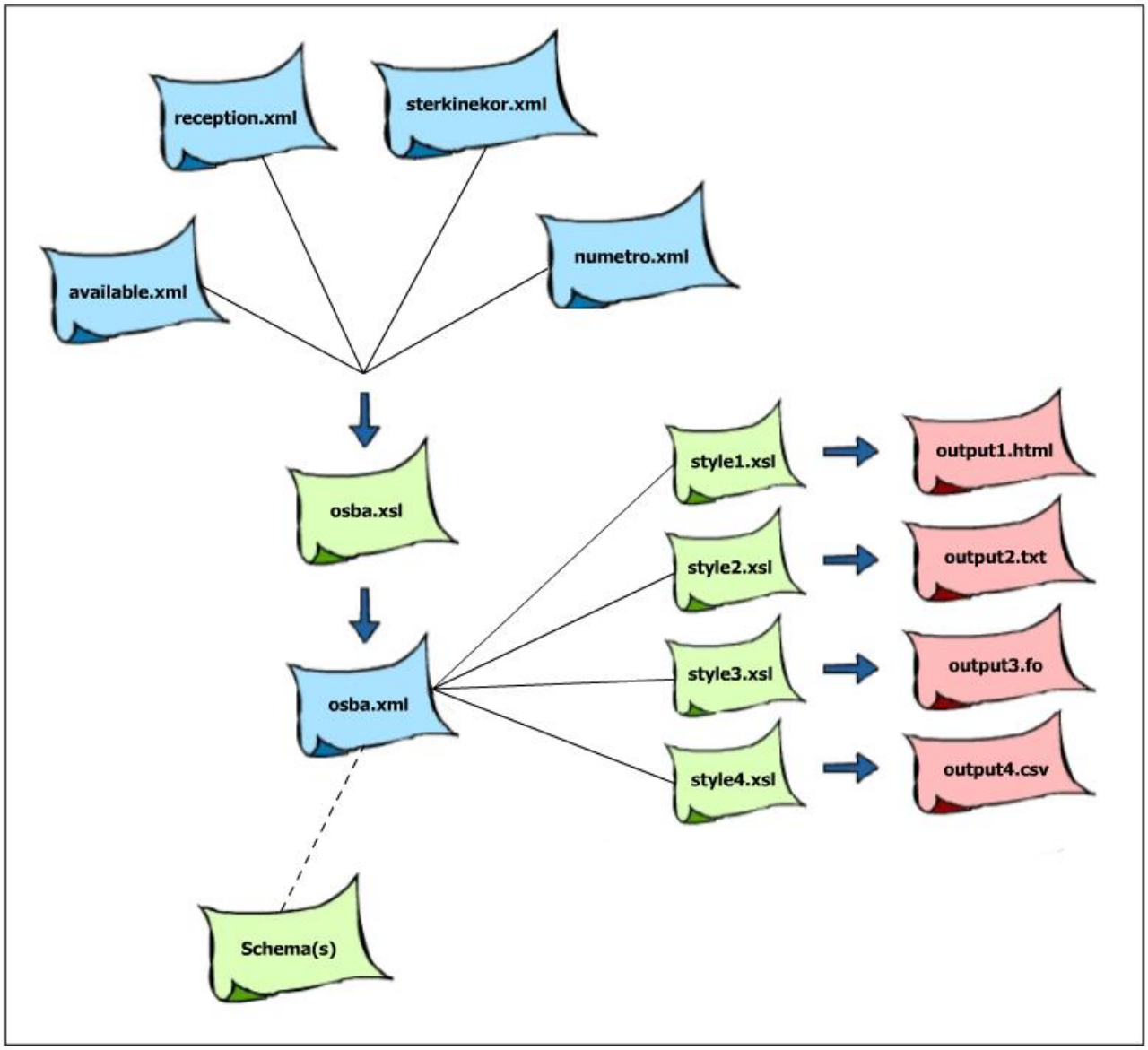
- The CSV file must contain the following information from **available releases only**:

    - Product title
    - Cover URL
    - Rating
    - Studio
    - Release date (format: dd/mm/yyyy)
    - Running time (format: *n* min)

- The CSV file must be **comma-delimited**. The first row must contain the headings of the columns, and subsequent rows will contain the data. Follow the USA/UK example in the above-mentioned Wikipedia article.

## Project Diagram

This diagram may help you visualise the process:




## Testing the Project

**This is the process you must follow to create your project's result documents (and also the process the marker will follow):**

1. Copy all provided files (XML sources, images, batch files, and MSXSL) into the same folder as the one containing your style sheets (i.e., no subfolders). Note that the image URLs in your project should be **relative** URLs referring to the current folder.

2. Run (double-click) "cmdLine.bat" to open a command prompt in the current folder.

3. In the command prompt, type: "combine" and press ENTER. This should result in "osba.xml". Note that the MSXSL command inside "combine.bat" may only have one source, and therefore uses "available.xml". You need to refer to the other sources inside the style sheet itself.

4. Now use an XML Schema validator to validate your "osba.xml" file according to your schema(s).

5. In the command prompt, type: "transform1" and press ENTER. This should result in "output1.html". Test this in the latest version of Firefox at a resolution of 1024x768 or above.

6. In the command prompt, type: "transform2" and press ENTER. This should result in "output2.txt". Test this in the Notepad text editor.

7. In the command prompt, type: "transform3" and press ENTER. This should result in "output3.fo". Use Apache FOP (available on the website under "Software") to format the FO document to PDF. Finally, test the PDF file using the latest version of Adobe Reader.

8. In the command prompt, type: "transform4" and press ENTER. This should result in "output4.csv". Test this using Microsoft Excel or OpenOffice.org Calc.

   (Note that some versions of Excel only read a CSV file properly if you open Excel **first**, then use File > Open to open it. When asked, choose "comma" as the delimiter.)

## Administrative Details

- You must work on this project **individually**.
- Plagiarism will not be tolerated; see the study guide in this regard. Generating code using unauthorised software will also count as plagiarism.
- Your project deadline is **Monday morning, 30 May 2011, 10:00**.
- You must submit your project using the "Project" link under "Assignments" on clickUP. You must submit the following files, compressed into a **ZIP** archive:
  - osba.xsl
  - osba.xml
  - Your schema(s) in XSD format
  - style1.xsl
  - style2.xsl
  - style3.xsl
  - style4.xsl
- **VERY IMPORTANT:** If your "osba.xsl" does not generate a master XML document ("osba.xml") from the source XML files successfully, you must create "osba.xml" manually, since the rest of the project depends on that file. Your project cannot be marked without this file!
- If you used any images apart from those provided, please include **only the extra images**. If you added a CSS style sheet to style your HTML page, include that too. All additional files will be placed in the same folder as your style sheets during marking, with no subfolders. Do not make use of special fonts that are not included in a native Windows installation, unless embedded inside an image.
- **DO NOT SUBMIT ANY OTHER FILES**. Do not upload any provided XML files or images and create a needlessly large submission. See the study guide regarding incorrect submission.
- Double-, triple-, quadruple-check your project archive before uploading, and after uploading, double-, triple-, quadruple-check the submission! Start to upload at least 30 minutes before the deadline to leave room for possible technical issues. If you make a mistake during submission and the deadline has not passed, use the "Take back submission" button on the submission page on clickUP to clear the submission, then resubmit.
- **DO NOT** e-mail the project to the lecturer unless you really and truly had no other choice. Provide a good reason in the e-mail to support your case.

- The marker will follow the steps under “Testing the Project” when marking, so ensure that everything works as it should using those steps.

*May it go well!*