

Report Project 1 : XML Schema Definition

Rémy Detobel & Nathan Liccardo

April 1, 2018

1 Introduction

This document aims at describing and explaining all the choices that we make for this project. As a reminder, the main purpose has been to create an XSD file containing a specific XML Schema Definition. This report will be divided into two parts : variable definitions and the XML-Schema structure.

2 Variables

This project use two kinds of variables : complex and simple. Most of the time, complex types are sequences of simple types. Firstly, let's define simple variables :

- *String* : title, publisher, abstract, edition, author, editor;
- *Integer* : volume, number, price, impact.

Regarding the complex types, they are used for defining year, genre and ISBN variables. As we must be able to declare a year using a tag or an attribute, we defined it into two parts. Firstly, we declared the format using simple types. Then, secondly, we defined an attribute and an element which are using this format. Genre and ISBN variables are both based on string restrictions. For the first variable, we restricted possible input strings (thriller, horror, sci/fi, romance, literature) using enumerations. For the second variable, we used patterns to restrict the input string format.

2.1 ISBN

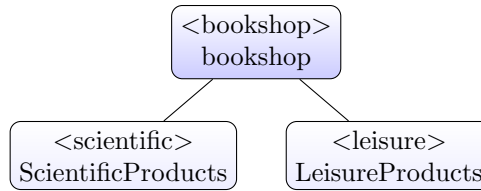
ISO define two ISBN formats : ISBN-10 and ISBN-13 (10 or 13 characters). In this project, we have arbitrarily choose to use ISBN-10. This standard split the code into four parts : a country (1 digit), an editor (2, 3 or 5 digits), a book number (6, 5 or 3 digits) and a verification code (1 digit). Has said before, we have been used patterns to check ISBN formats. Those patterns are defined as follows :

- $\backslash d\{1\}-\backslash d\{5\}-\backslash d\{3\}-\backslash d\{1\}$
- $\backslash d\{1\}-\backslash d\{3\}-\backslash d\{5\}-\backslash d\{1\}$
- $\backslash d\{1\}-\backslash d\{2\}-\backslash d\{6\}-\backslash d\{1\}$

Which means that we can only use digits, in a structured forms and separated by dashes.

3 Structure of the schema

We were asked to write an XSD file defining a book shop. This shop had to be composed by scientific leisure products. This first relation is represented by the tree below. Lighter-coloured leafs represent non-mandatory elements.



Each nodes contains a tag (XML) and a type name. Firstly, we can observe that **bookshop** have two children : **scientific** and **leisure** (see below). **Bookshop** may be empty, composed by one of the two element or by both. Finally, we obtain the below XML structure :

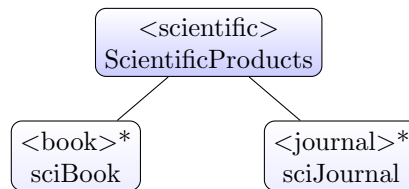
```

<bookshop>
  <scientific>
    ...
  </scientific>
  <leisure>
    ...
  </leisure>
</bookshop>

```

3.1 Scientific products

Scientific products are separated into two sub-categories : scientific books and scientific journals. For both of them, we can have zero or more occurrences. This property is represented using the asterisk in the following tree :



These two new type (sciBook and sciJournal) are a little more complex than what we have seen so far. As **bookshop**, **scientific** can be empty or contain only one of the two types. Now we obtain the below XML structure :

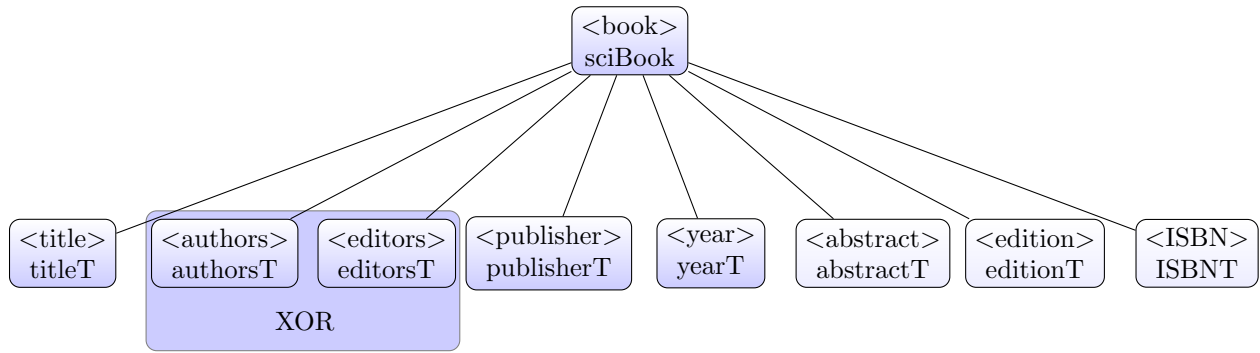
```

<bookshop>
  <scientific>
    <book>
      ...
    </book>
    <book>
      ...
    </book>
    <journal>
      ...
    </journal>
    ...
  </scientific>
  ...
</bookshop>

```

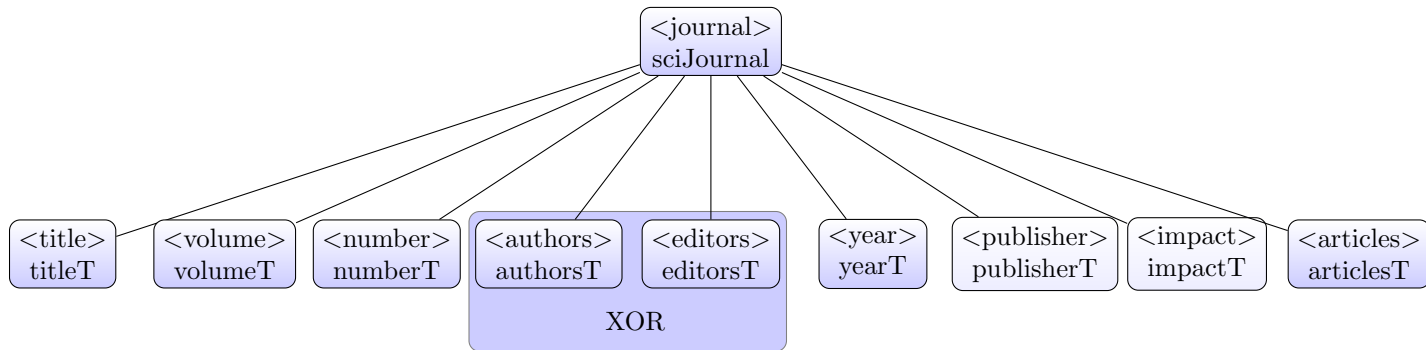
3.1.1 Scientific book

Scientific books are defined using simple types (previously defined) and two new complex types : **<authors>** and **<editors>**. Both contains list of elements (respectively “authorT” and “editorT”) and cannot appears at the same time. Please note also that the three last elements are optional.

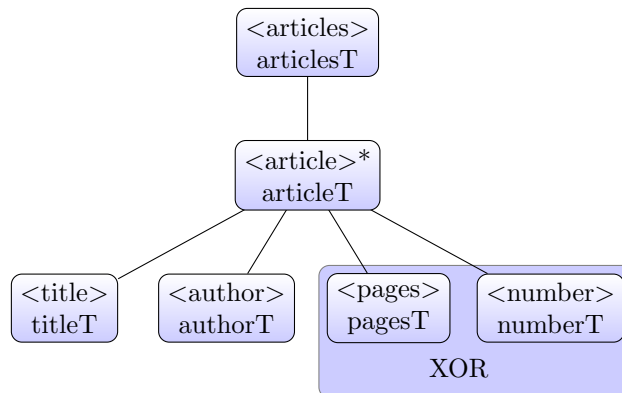


3.1.2 Scientific journal

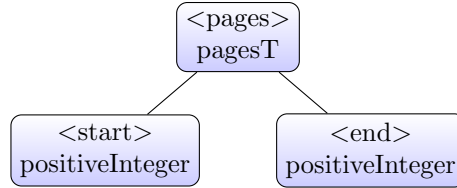
Scientific journals are structured as same as for scientific books. The difference lay in the used types (all are already defined excepted **articles**). Regarding to **author** and **editor** types, they are defined as for scientific books.



Articles A scientific journal must have a list of articles. Therefore, the **articles** tag contains at least one **article** tag which itself contains attributes. This can be represented as below :



Like **authors** and **editors**, **pages** and **number** could not appear at the same time. A **pages** element must have a start and end page (represented using *Integer* types). This relation is represented as follows :

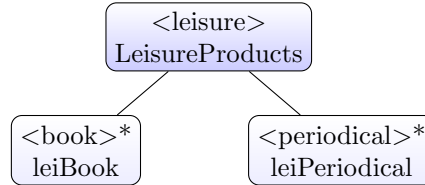


Impact As indicated previously (see section 2), the “impact” type is an integer. This integer is calculated according two factors : the number of citations (in one year) of articles published in that journal (within the two last years) and the total of articles published in that journal during the same two years. Finally, to obtain the “impact” value, we must divide the first factor by the second one. Therefore, impact value will be different each year. This implies that the impact is characterised by a “year” attribute. Below is the final XML structure :

```
<impact year="XXXX"> ... </impact>
```

3.2 Leisure products

Leisure products are separated into two parts : leisure books and leisure periodicals. The following relation has the same structure as for scientific products :



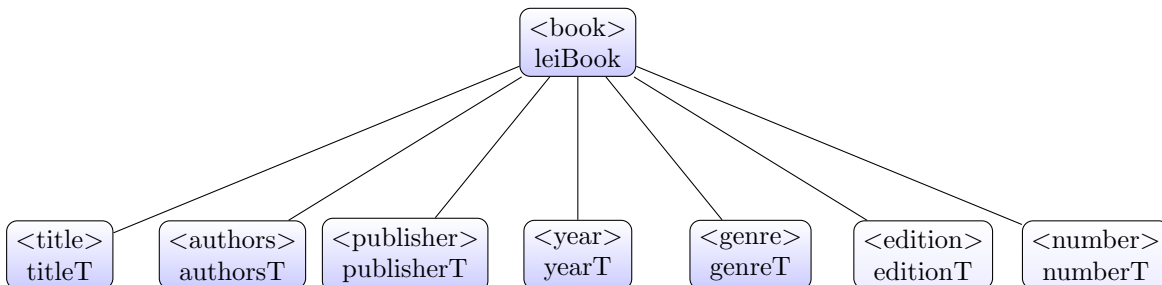
Once again, we can construct the XML code :

```

<bookshop>
...
  <leisure>
    <book>
      ...
    </book>
    <periodical>
      ...
    </periodical>
    <periodical>
      ...
    </periodical>
  ...
</leisure>
</bookshop>
  
```

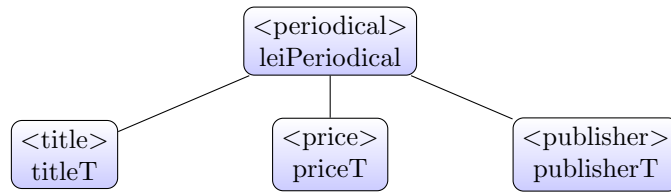
3.2.1 Leisure Book

Leisure books uses exactly the same structure as previous elements. In addition, all the used types have already been defined.



3.2.2 Leisure Periodicals

Leisure periodicals uses exactly the same structure as previous elements. Once again, every used types have already been defined.



A Relation type

- $\text{bookshop} \rightarrow (\text{scientific}, \text{ScientificProducts}), (\text{leisure}, \text{LeisureProducts})$
- $\text{ScientificProducts} \rightarrow (\text{book}, \text{sciBook})^*, (\text{journal}, \text{sciJournal})^*$
- $\text{LeisureProducts} \rightarrow (\text{book}, \text{leiBook})^*, (\text{periodical}, \text{leiPeriodical})^*$
- $\text{sciBook} \rightarrow (\text{title}, \text{titleT}), (\text{authors}, \text{authorsT})^+ \mid (\text{editors}, \text{editorsT})^+, (\text{publisher}, \text{publisherT}), (\text{year}, \text{yearT}), (\text{abstract}, \text{abstractT})?, (\text{edition}, \text{editionT})?, (\text{ISBN}, \text{ISBNNT})?$
- $\text{sciJournal} \rightarrow (\text{title}, \text{titleT}), (\text{volume}, \text{volumeT}), (\text{number}, \text{numberT}), (\text{authors}, \text{authorsT})^+ \mid (\text{editors}, \text{editorsT})^+, (\text{year}, \text{yearT}), (\text{publisher}, \text{publisherT})?, (\text{impact}, \text{impactT})?, (\text{articles}, \text{articlesT})$
- $\text{articlesT} \rightarrow (\text{article}, \text{articleT})^+$
- $\text{articleT} \rightarrow (\text{title}, \text{titleT}), (\text{author}, \text{authorT}), (\text{pages}, \text{pagesT}) \mid (\text{number}, \text{numberT})$
- $\text{pagesT} \rightarrow (\text{start}, \text{positiveInteger}), (\text{end}, \text{positiveInteger})$
- $\text{leiBook} \rightarrow (\text{title}, \text{titleT}), (\text{authors}, \text{authorsT}), (\text{publisher}, \text{publisherT}), (\text{genre}, \text{genreT}), (\text{edition}, \text{editionT})?, (\text{number}, \text{numberT})?$
- $\text{leiPeriodical} \rightarrow (\text{title}, \text{titleT}), (\text{price}, \text{priceT}), (\text{publisher}, \text{publisherT})$