Software Architectures
Assignment 3 : Service Oriented Architectures - Web Services

Arnaud Rosette, Simon Picard

April 20, 2015

1 Exercise 1 : BPEL Processes

1.1 BPEL

To parallelize the invokation to the web services, the order of opperation must be modified, the two "AssignSearchRequest" have to be follow each other in order to be able to performe the invokations one after the other. Once this is done, BPEL provides a flow construction wich allows task to be executed simultaniously. There was a small error in the "AssigneSearchRequest", to copy the query, the orignal process used "string($request.bodyquery)" but the correct form is "string($request.bodytns:query)".

Two sections of the process are parallelized, the one where the response page and the search requests are set up and the part where the calls to the library are actually done. It is clear that theses calls could not be perfomed earlier because they need their parameter to be set which is done in the preparation of the search requests. Therefore, the "AssignSearchRequest" are data dependencies to the invokations. And, of course, the next part of the process, where the results of the library searchs are written in the response and then the response is sent, is sequential.

In the flow structure, links can be used to state some control dependencies, in our case we could link the part where parameter for a call are set to the call itself, stating that tasks can be execute simultaniously but these two operations have to be performed in a certain order.

1.2 WSDL

Here ane example of result of the service.

```
<searchBooksResponse xmlns="urn:soft.vub.ac.be/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:
    <book xmlns="">
        <author>Jordan Belfort</author>
        <year>2008</year>
        <isbn>606259090</isbn>
        <language>en</language>
        <publisher>Hodder Paperback</publisher>
        <title>The Wolf of Wall Street</title>
    </book>
</searchBooksResponse>
<searchForBooksReturn xmlns="http://library.be" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xs
    <author>James, E. L.</author>
    <date>3914-01-10T23:00:00.000Z</date>
    <isbn>0345803485</isbn>
    <language>English</language>
    <publisher>Vintage Books</publisher>
    <title>50 Shades of Grey</title>
</searchForBooksReturn>
```

The responses share balise author, isbn, language, publisher, title and the xmlns informations. For the date, the Soft Library return only the year while the National Library gives a full date. For the arcitecture, the Soft Library begin with a searchBooksResponse balise in which the xmlns informations are

stated and then there is a nested entry for each book, in the National Library ther is no containing balise, each book contains the xmlns values.

Once the library returned the results, the datas are parsed using the "bpel:doXslTransform", a function which allow the process to use a xsl steelsheet to transform the datas. the mainnadvantage is that we could add other library to the protocol and use still use this way of parsing, one would just have to modify a bit the xsl file and add two parameters in the doXslTransform function. Once this transform is done, the content of the result is copied to books balise in the response form.