

Software Architectures

Assignment 3 : Service Oriented Architectures - Web Services

Arnaud Rosette, Simon Picard

April 23, 2015

1 Exercise 1 : BPEL Processes

1.1 BPEL

The order of operations must be modified in order to parallelize the invocation of the web services. The two "AssignSearchRequest" have to be executed one after the other to be able to perform the invocations in parallel. Once this is done, BPEL provides a flow construction which allows tasks to be executed simultaneously. There was a small error in the "AssignSearchRequest", to copy the query, the original process used "string(\$request.bodyquery)" but the correct form is "string(\$request.bodytns:query)".

There are two sections of the main process which are executed in parallel, one where the response page and the search requests are set up and another where the calls to the library are actually done. It is clear that these calls could not be performed earlier because they need their parameter to be set which is done in the preparation of the search requests. Therefore, the "AssignSearchRequest" share data dependencies with the invocations of the web services. And, of course, the next part of the process, where the results of the library searches 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 parts where parameter for a call are set to the call itself, stating that tasks can be executed simultaneously but these two operations have to be performed in a certain order.

1.2 Querying web services

Here is an example of a result of the service.

```
<searchBooksResponse xmlns="urn:soft.vub.ac.be/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <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:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <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 attributes such as author, isbn, language, publisher, title and the xmlns information. For the date, the Soft Library return only the year while the National Library gives a full date. For the architecture, the Soft Library begin with a searchBooksResponse tag in which the xmlns information is stated and then there is a nested entry for each book, in the National Library there is no containing tag, each book contain the xmlns values.

Once the library returned the results, the data is parsed using the "bpel:doXslTransform". It is a function which allow the process to use a xsl steelsheet to transform the data. the main advantage is that we could add other library to the protocol and 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 tag in the response form.