Project for the *Software Service Engineering*[1] Course (a.y. 2015/2016)

The project consists of designing and analysing an executable specification of a (meaningful) orchestration of three existing remote Web services A, B and C. Each project instance must satisfy the following requirements:

- The invoked services must include a SOAP service and a RESTful service.
- None of the services used in the lab can be used.
- (Part of) the results of the invocation of service B must be used to invoke service C.
- Service A must be invoked in parallel to the invocations of services B and C.
- Service A must be invoked asynchronously by invoking a process that acts as a proxy.
  - The orchestrator must throw a fault `to_fault` if it receives no reply from the proxy within a given time. The throwing of `to_fault` must terminate the invocations of B and C (if those had not been already executed).
  - The orchestrator must also throw a fault `reply_fault` if the reply received from the proxy does not satisfy some given condition. The throwing of `reply_fault` must NOT terminate the invocations of B and C.

Each delivered project instance must include:

(a) A WS-BPEL process P implementing the orchestrator and running on OpenESB v2.3.x.

(b) A sound workflow net correctly modelling (the control flow of) P. The net must properly model activity termination when a fault is thrown. The labels used to name transitions must be meaningful and the net must be easy to read.

Instructions for project delivery

Each project must be delivered to the Instructor via email with a single message with subject "SSE PROJECT DELIVERY". The email should have an attached zipped folder named *studentSurname* containing:

- All the files needed to deploy, run and analyse the orchestrator, and a PDF *readme* file clearly explaining the content of the folder.
- A PDF file containing a project report. The report must be at most 10 pages (with font size ≥ 10) and it must contain the following sections:

1. Introduction. This section should clearly describe –with the help of a figure- the chosen orchestration. In particular which are the inputs and outputs of the orchestrator and of the service invocations, and how such inputs and outputs relate one another. The actual addresses of the employed remote services must be explicitly mentioned.
2. WS-BPEL implementation. *4.1 WS-BPEL processes.* This section must clearly describe the main WS-BPEL process P, and it must contain readable figures illustrating (parts of) such processes. *4.2 Tests.* This section must describe which tests were successfully executed. Such tests must include the case in which no fault was thrown, the case in which `to_fault` was thrown, and the case in which `reply_fault` was thrown.
3. Analysis of the WS-BPEL specification. This section should describe the workflow net modelling (the control flow of) P.

Both the project report and the readme file must be written in English, and they will be object of evaluation. Failure to meet any of the requirements set by the above specification or failure to implement a simple change to the delivered project (during the project discussion) will cause the rejection of the project delivery.

A list of frequently asked questions will be maintained by the Instructor and made accessible from the course Web page. The answers contained in such a list are to be considered a true integration of this project specification.

[1] Students of the *Software Servic*es course must deliver only (b).