Distributed Programming II

A.Y. 2014/15

Assignment n. 4 – part a)

All the material needed for this assignment is included in the *.zip* archive where you have found this file. Please extract the archive to an empty directory (that will be called *`[root]`*) where you will work.

This assignment is about the design of a set of web services for the FDS system. The web services have to be designed according to the following specifications.

- The system includes the following client hosts/processes:
  - **Check-in terminals** are clients that can only perform check-in operations
  - **Boarding terminals** are clients that can only perform board management operations
  - **Operator consoles** are general purpose clients that can perform all the operations (i.e. check-in operations, boarding management operations and other operations).
- The web services to be designed must provide the following functionalities:
  1. **Check-in operations**: a check-in operation assigns a seat to one of the passengers registered for a flight instance and returns the boarding pass information for that passenger (i.e. the assigned seat, the flight instance gate if already available, and the delay if any).
  2. **Board management operations**: start boarding for a flight instance (i.e. change the status of a flight instance from CHECKINGIN to BOARDING), register single passengers as boarded for the flight instance on which boarding has started, and get the list of passengers currently boarded on the flight instance on which boarding has started. The list must include passenger name and seat. Only passengers that have already been checked in can be boarded
  3. **Other operations**: cancel a flight instance (i.e. change its status to CANCELLED), change information about boarding gate and delay of a flight instance, read all the available information about flights, flight instances, and passengers. This last functionality must provide read access to the same information that can be accessed through the Java interfaces in the package `it.polito.dp2.FDS` (the one used for the first two assignments). Note the Java interfaces in the package `it.polito.dp2.FDS` are object-oriented and have been designed for local access, while the web service interface(s) should be service-oriented, and for remote access.

Assuming the FDS system has to be developed using standard (SOAP) web services, design the web service(s) according to the specifications given above. Specify the designed web services inside two WSDL documents named respectively `FDSInfo.wsdl` (including the web service(s) that provide read access to all the available information) and `FDSControl.wsdl` (including the web service(s) that provide the other functionalities). Save both files under *`[root]/wsdl/`*.

The WSDL files can reference other WSDL files stored in the same directory, and type definitions can be saved separately in a schema file with the same name as the main WSDL file and the xsd extension (for example, FDSInfo.xsd), stored in the same directory.

The web services must be designed so as to be robust and so as to make the completion of the specified operations both efficient and easy to be used by clients.

The WSDL documents must include comments (`documentation` elements) that explain the semantics of the operations (so that users have all the information that is necessary for using them) and that briefly explain the main design choices and assumptions made.

When designing the system, consider that it should be able to manage a multiplicity of airports, with a total of thousands of flights. When designing read access to all the information available, consider that this information may include data related to both the past and the future, and it can be related to a long time period (even several years).

Correctness verification

In order to be acceptable for examination, the produced WSDL documents must be valid. This can be checked by means of the Eclipse WSDL validator.

Submission format

The produced WSDL documents will be submitted along with Part b) of Assignment 4, for which instructions will be given later on.