Distributed Programming II

A.Y. 2015/16

Assignment n. 1 – part a)

The material for this assignment is in the *.zip* archive where you have found this file. Extract the archive to an empty directory that you will use as your working area and that we will call `[root].`

DP2-WF is a (simplified) distributed system that manages *workflows*. The workflow concept and terminology used in the DP2 assignments is explained in the document `[root]/intro.pdf`.

The Java interfaces defined in the package `it.polito.dp2.WF` (available under `[root]/doc` and at https://pad.polito.it:8080/enginframe/dp2/assignments/lab1/doc/index.html) give read-only access to some information about a set of workflows and related processes. The javadoc of the interfaces documents the kind of information that can be retrieved. The main interface, from which all the information can be accessed, is `WorkflowMonitor`. The methods in this interface can be used to get the available information about known workflows (`getWorkflows`), and processes (`getProcesses`). The latter two methods give lists of `WorkflowReader` and `ProcessReader` interfaces respectively, by which all the available information about workflows and processes can be obtained. In the file `[root]/info.pdf` it is possible to see two sample workflows with related processes.

The solution to part a) will be submitted along with the solution to part b).

Assignment description

1. Design an *XML* application that can be used to store all the information that can be retrieved by using the *Java* interfaces defined in the package `it.polito.dp2.WF` starting from interface `WorkflowMonitor`. The XML format must be such that all the data that can be retrieved using such interfaces can also be obtained from the *XML* document, and redundancies should be avoided. The *XML* application must be specified by means of a *DTD*, which must be saved in the file `[root]/dtd/ wfInfo.dtd`.
2. Write a short documentation of your design choices about the DTD (max 1 page) and save it as an ASCII file in `[root]/dtd/doc.txt`.
3. Write a valid *XML* file that references the designed *DTD locally*. The file must be saved as `[root]/dtd/wfInfo.xml`.

Correctness verification

Before submitting your files, please verify their correctness. The submitted solution must at least satisfy the following requirements, in order to be considered acceptable:

- the file `wfInfo.dtd` must be syntactically correct;
- the file `wfInfo.xml` must be valid and must reference the DTD stored in the same directory, in file `wfInfo.dtd`;
- the file `doc.txt` must exist.

The validity of the XML file can be checked by any XML validation program. For example, it can be checked by the Eclipse validate command or by running the `DomParseV` program (distributed in the *XML* examples bundle on the course site).