

SOFTWARE ENGINEERING 2015 – TASK 1

Project name: Parameter Management System

This project is based on the existing code, which has been developed following the general description given below. Please pay attention to the following issues:

- analyze the correctness of the code (style, architecture, integration, and run-time behavior),
- design and create the testing framework (a program that can be run with different parameter files to verify the correctness of the Parameter Management System),
- review the functionality with regards to the description (implemented, missing),
- identify the bugs and problems, and fix the critical ones,
- change the program operation to files and folders (and possibly come config files) instead of using the Access database.

Background

Research aimed at creating new algorithms usually includes tuning parameters for various algorithms. This is a very important task and it is very helpful to keep all the parameters in a single file, e.g. in an XML file. However, managing multiple parameter files may be quite challenging as sometimes dozens or hundreds of them are generated. Hence, there is a need for a system that would support such XML files management.

Task description

The system will work with XML files of a defined structure (however, this should be configurable somehow). An example is given below:

```
<PARAMETERS>
  <GROUP NAME="ParamGroupName" HINT="Text (group desc)">
    <PARAM NAME="ParamName" HINT="Description..." TYPE="Int" VALUE="1" LEVEL="3"></PARAM>
    <PARAM NAME="2ndParam" HINT="Description..." TYPE="Double" VALUE="0.000000" LEVEL="3"></PARAM>
    ...
  </GROUP>
  <GROUP NAME="ParamGroupName" HINT="Text (group desc)">
    ...
  </GROUP>
  ...
</PARAMETERS>
```

The groups of parameters can be nested. Allowed values of the parameters:

- Name – any text without blank characters.
- Hint – any text.
- Type – Int, Float, Double, Bool, Text.
- Level – 1, 2 or 3.

The system should provide the following:

1. Files storage. The files should be stored in a database.
 - a. Adding new XMLs. This should be possible from GUI and from a command line (using a script).
 - b. Tagging and time stamp. A new XML may be assigned with a tag, regardless of the adding procedure. It should be possible to add a tag (or tags) to existing files as well.

- c. Deleting an XML.
 - d. Searching the files based on parameter value (or range), time stamps and tags.
- 2. Content-based XML comparison. GUI should visualize the differences in the values of the parameters (not just differences in text).
 - a. Comparing two XMLs.
 - b. Comparing multiple XMLs. The differences between the parameters should be shown at different levels (e.g.:
 - i. parameter name "Alpha", values range 2.3-5.4,
 - ii. parameter name "Alpha", values range 2.3-5.4 (11 values 2.3, one value 5.4),
 - iii. parameter name "Alpha", full list of values)
- 3. Planning and executing experiments.
 - a. Generation of XMLs containing parameters of defined values. The values can be defined by range and step or by giving a full set of possible values. It may be helpful to see a list of possible combinations before the XMLs are generated.
 - b. A generated XML should be stored to a given location as a file. After that a given file script (*.bat) should be executed.
 - c. It should be possible to notify the system that a script has been finished, so that the next XML can be stored and new experiment started (it may be required that a batch file contains a certain, defined command). Only one experiment should run at a time.
 - d. Logging the execution. XML and script start (and notification) time stamps should be stored somewhere and displayed to the user).