OATS Script Generator (Placeholder Name)

Responsible: Marina, Maurício

Date Redacted: 19/01/2015

Proposal: The creation of a new Script Generator component for OATS based on the current OATS script generator component for Web applications, for the testing of Oracle applications.

Recommended starting point: It is unlikely that extending Ogma to be usable as an universal compiler would cause compatibility issues. Regardless, the use of Ogma's basic format is advised.

Deadlines:

10/02/2015 -> Acquisition of a testing script for an Oracle Application to use as a base for development and analysis of the tags required by this script.

PLeTs UI Overhaul (Placeholder Name)

Responsible: Juliana

Date Redacted: 19/01/2015

Proposal: Either the creation of a new PLeTs GUI or the improvement of the current GUI in order to update its functionalities in accordance with the current version of the PLA (v 3.0). The components of the PLA that refer to Performance or Structural Testing (listed below) do need necessarily need to be covered by the new project, but it would be advantageous to have them in mind when designing it. The primary goal is to make the GUI self-explanatory, so as to minimize the need for an instruction manual and enable users to learn it without the need to consult the design and development team (Coc-Dell).

New Components: Either a new GUI component to replace the current Main Window, or an update of the Main Window itself.

Performance or Structural Testing Components: Performance Validator, LR, VS, JMeter, JMeter Exec, VS Exec, LR Exec.

Observation: The result of this project will be a first version of the GUI for PLeTs and does not necessarily have to cover variability. It is acceptable to have several GUI components and include a GUI Factory to the PLA, where each of these components will refer to a single product configuration. Coc-Dell is currently uncertain as to whether we have interest in researching UI variation management.

Deadlines:

10/02/2015 -> Either one working new GUI for Functional Testing with variability covering both OATS and MTM, or two new GUIs for Functional Testing, one covering OATS and another covering MTM.

.NET Web Integration (Placeholder Name)

Responsible: Cristiano

Date Redacted: 19/01/2015

Proposal: The study of methods of integrating a .NET-based software project into the web. We hope to create web-based application fronts for Plug with minimal modifications to the existing components. We must have an evaluation of the options available to us, a listing of the advantages and disadvantages of each of them. Particular focus must be given to the following points:

- How difficult would this particular approach be for the development team to learn?
- How much refactoring and/or re-implementation would be necessary to the existing components with this approach?
- How difficult would it be to give maintenance to this approach?
- What precedent is there to the use of this approach?

All of these points are expected to be given as rough estimates, and will not be considered final regardless of the decision taken by the design team.

Recommended starting points: WPF with WCF; ASP.NET MVC Application; Silverlight.

Glossary:

WPF: Windows Presentation Foundation

WCF: Windows Communication Foundation

Deadlines:

10/02/2015 -> Complete report on the available technologies researched based on the questions from the project proposal. Ideally, small examples of functioning applications in each of the researched technologies (Hello World button would suffice).

Multi-language SPL Support (Placeholder Name)

Responsible: Édio

Date Redacted: 19/01/2015

Proposal: The implementation of a component for PLeTs in any language that is not natively supported by the .NET framework. By creating a component outside the .NET framework and linking it successfully with PLeTs, we become open to the possibility of working with other programming languages.

Recommended starting point: The suggested starting point is the creation of a parser in java, given the broad knowledge of the language within our environment and the relative simplicity and autonomy of this phase of PLeTs.

Deadlines:

10/02/2015 -> Prototype of a java component connected to the PLeTs Control Unit. Functionalities related to it do not have to be fully implemented, that is, bugs not related to the communication between components is acceptable.

Second PLeTs Modeling Structure (Placeholder Name)

Responsible:

Date Redacted: 19/01/2015

Proposal: The study of a second modeling structure to be implemented in PLeTs alongside UML. From the point of view of SPL research, we are currently somewhat limited in that we only have one modeling structure to work from, whereas we have multiple structures for all others phases. It is difficult to conceive and explain certain aspects of SPL design due to this limitation. This new modeling structure does not necessarily need to be given thorough maintenance and support, but would be helpful in the analysis of the SPL.

New Components: A new component will be necessary to represent the new modeling structure. Another will be necessary for, at least, the input parsing of this structure into PLeTs. And output parsing component would be ideal, but is not necessary. Finally, a converter between the new modeling structure and at least one of the intermediate Sequence Generation structures will be necessary to validate the project.

Intermediate Sequence Generation Structures: Graph, FSM, VFSM.

Recommended Starting Points: Splot, Clafer

Second UML Modeling Software Support (Placeholder Name)

Responsible:

Date Redacted: 19/01/2015

Proposal: The study of a second UML modeling software to be supported by PLeTs by the presence of an input/output parser. The current reliance on Astah as the only supported UML modeling software for PLeTs is a severe limitation to the study of SPL aspects, as well as being a limitation to the adoption of PLeTs in external environments. To have a second parser would largely facilitate the study of the application of design patterns in variability solving, as well as serving as a proof of concept as to the versatility of PLeTs.

New Components: An input/output parser for a new UML modeling software. This parser must necessarily give an UML output equal to the UML output of the Astah parser, so as to be compatible with the other phases of PLeTs without the need for refactoring.

Observations: The selection of the new tool for which to be given support is not necessarily important, but prior research into its adoption rates would be ideal. That is, creating a parser for a tool that is actually used by the industry would be more pragmatically advantageous than creating a parser for an obscure tool.