Visual Construction System of Interfaces

Escola Superior de Tecnologia e Gestão de Beja

Cláudia Oliveira, N.º 3459 – Cláudio Pedro, N.º 3805 – Nuno Coelho, N.º 3938

June of 2007

Abstract

This document represents an approach to what was done by the students during the development of a visual construction system of interfaces.

I – INTRODUCTION

This project had as objective the development of a graphical tool that supports the construction of interfaces composed by elements of the graphical toolkit qooxdoo [1]. It is intended, with this tool, to diminish the time expense for the programmers in the construction of the interfaces, facilitating still the construction to them, of the same ones. The development of graphical interfaces in text-mode takes too much time and makes it harder for a programmer to get the job done, given the clear difficulties that he will have to face. There are nowadays, innumerable tools of IDE type that make it easier to build well-structured and good-looking interfaces, interactively and fast. In the development of this tool, it was appealed to the graphical toolkit Qt (Dalheimer, 2002) [2], which presents two advantages of extreme relevance. It allows drag & drop and its graphical controls possess layouts similar to the ones of controls with the same function in the qooxdoo toolkit.

II – METHODOLOGY

In the development of this tool, it was adopted the waterfall model [3], which is constituted by five distinct phases. *Analysis*, in which analogous tools were studied and the characteristics to implement where identified. *Drawing*, where the modules of the implementation, its characteristics and the foreseen times of development, where identified. *Implementation*, during which the modules identified in the previous phase, were implemented. *Tests*, which was useful to verify that the requirements detected during the analysis phase had been implemented, and *Maintenance*, which was not considered during the accomplishment of this project.

III – TECHNOLOGIES USED

In the development of the application, it was used the programming language Python, the graphical *toolkit* Qt, the codification languages YAML and HTML, the programming language *JavaScript*, the AJAX technologies [4] and the qooxdoo framework.

IV – WORK DEVELOPED

The development of the application was initiated searching on the Web for analogous tools and studying those. Then the Use Cases diagram was elaborated, its templates where filled and three possible scenes of use for the application were described.

In a second phase, the Classes diagram and the State Transitions diagram were elaborated, and the low-fidelity prototypes of graphical interfaces of the application were drawn. These prototypes included the window of the application as well as dialog windows to confirm actions. During the development of these prototypes, a heuristic evaluation [5] to the same, based on the 10 heuristics of Nielsen was done.

With the specification of the requirements and the modeling of the system done, the implementation phase arrived. This phase consisted in the codification of the following four modules of implementation:

- Collect of information about the visual controls to be available to the user to build interfaces and selection of the same. Another study was still done, to define which controls were similar between the controls of qooxdoo and Qt;
- Construction of the graphical interface of the application and implementation of

mechanisms to interact with the available visual controls. The graphical interface was developed based on the low-fidelity prototypes drawn during the modeling of the system. Each visual control is represented by a specific class that provides the interaction with the application;

- Construction of a HTML code generator that will include a portion of JavaScript code, where, through qooxdoo commands, the interface will be described. The HTML code generator is the "engine" that provides the user with the possibility of previewing the interface, in a browser.
- Construction of a YAML code interpreter and generator that allows the storage and loading of interfaces and templates created by the user. In the developed application the user will have the opportunity to create interfaces and templates that he might wish to store in a storage unit of its choice, to further visualize or modify. The extension of the files that store interfaces is ".ymli" and the extension of the files that store templates is ".ymlt".

V – CONCLUSION

This project allowed above all a deepening of knowledge at the level of applications that make the development of graphical interfaces easier. It also allowed the application of knowledge previously acquired in various classes of the course. In short, this project was extremely competitive and interesting due to the diverse knowledge that it required being that it will be equally interesting to follow the evolution process of the developed application, as well as, the increase of its potentialities and functionalities.

VI – BIBLIOGRAPHY

[1] *qooxdoo*. (2006, December 22). [Online]. Available at http://qooxdoo.org

[2] Matthias Kalle Dalheimer, *Programming with Qt, 2nd*

Edition, O'Reilly, January 2002, chapter 1

[3] *Waterfall model*. (2007, Janeiro 28). [Online]. Available at http://en.wikipedia.org/wiki/Waterfall_development

[4] *Ajax (programação)*. (2007, January 14). [Online]. Available at http://pt.wikipedia.org/wiki/AJAX_%28programa%C3%A7%C3%A3o%29

[5] *Heuristic Evaluation*. (2005). [Online]. Available at http://www.useit.com/papers/heuristic