

Bolsas de Investigação DEI/CISUC (LEI/LDM)

Outubro – Dezembro 2013

Proposta #1 (Preferência)

Título: A framework for testing web services interoperability

Orientador: Nuno Laranjeiro; Marco Vieira

Enquadramento

The web services technology was designed with Interoperability as one of its major goals. Despite being a mature technology, issues in the definition of the specifications and in the development of the platforms that support web services frequently prevent this goal from being fully achieved. In fact, these issues can be the source of problems in the communication between service consumers and providers and lead to failures and potentially huge operational, financial and reputation losses. Although some actions have been taken to correct interoperability issues (such as the actions taken by the Web Services Interoperability Organization) there is still no practical way to measure the interoperability of a web service, which is precisely the goal of this work.

Objectivos

The goal of this work is to create an approach and an experimental platform to allow service developers to assess the interoperability of their services from a practical perspective. The approach will consist of running a large set of client-side code artifact generator tools (e.g., wsimport, wsconsume) and collecting the success/error/warning information available by each tool.

Plano de Trabalhos

[Some tasks overlap]

T1 (October): Knowledge transfer on web services development.

T2 (October): Design of a set of standard web services (one operation each and one argument for operation) to be used to validate the approach.

T3 (October-November): Design and implementation of the experimental platform. This includes:

- Definition of a preliminary set of generic rules for integrating two major client-side code generator tools;
- Design and implementation of the platform for integrating code generation tools (in general);
- Practical and preliminary analysis of the tools output (to be considered for integration in the tests results);
- Execution of tests.

T4 (December): Integration of more code-generation tools and tuning the design of the experimental platform.

T5 (December): Execution of tests and writing a final report.

Proposta #2

Título: wsrbench v2: an online tool for web services robustness testing

Orientador: Nuno Laranjeiro; Marco Vieira

Enquadramento

Previous research has proposed that web services robustness testing should be based on a set of robustness tests (i.e., invalid web services call parameters) that are applied at runtime in order to discover both programming and design errors. Web services can then be classified based on the failure modes observed. **wsrbench** (<http://wsrbench.dei.uc.pt>) is an on-line web application that provides an easy interface for robustness testing of simple web services and has had impact in both the research and in the industry communities. Despite this, it still holds limitations regarding the processing of complex service interfaces and the number of data types that it can currently test. This limits the tool's adoption by developers that, many times, want to test complex services.

Objectivos

This work comprises two goals. The first goal is to extend the interface (wsdl) reading capabilities of wsrbench to allow processing of more complex service interfaces. The second goal is to extend the application's current fault model and implement robustness tests for more data types, as wsrbench is currently limited to test operations that handle numbers and strings.

Plano de Trabalhos

[Some tasks overlap]

T1 (October): Knowledge transfer on web services robustness and on wsrbench's architecture and API. Definition of a set of services with complex interfaces.

T2 (October-November): Changing the application to be able to read and process complex interfaces. This implies re-implementing the current reading module of the application and then mapping the changes to the data and presentation layer.

T3 (December): Extension of the application's fault model to be able to test more data types (e.g., Booleans, dates, lists).

T4 (December): Execution of tests and writing a final report.